Recognizing Berge Graphs

Maria Chudnovsky, Princeton University, Princeton, NJ

Gérard Cornuéjols[1], Carnegie Mellon University, Pittsburgh, PA, and
Faculté des Sciences de Luminy, Marseille, France

Xinming Liu, Carnegie Mellon University, Pittsburgh, PA

Paul Seymour[2], Princeton University, Princeton, NJ

Kristina Vušković[3], University of Leeds, Leeds, UK

November 15, 2002; revised May 20, 2004

Abstract

A graph is *Berge* if no induced subgraph of G is an odd cycle of length at least five or the complement of one. In this paper we give an algorithm to test if a graph G is Berge, with running time $O(|V(G)|^9)$. This is independent of the recent proof of the strong perfect graph conjecture.

1 Introduction

A *hole* in G is an induced subgraph of G that is a cycle of length at least four, and it is *odd* or *even* if it has odd (or even, respectively) length. A graph is *Berge* if G and its complement both have no odd hole. (The *complement* $\overline{G}$ of G is the graph with vertex set $V(G)$, in which two distinct vertices are adjacent if and only if they are nonadjacent in G.) It has been an open question whether there is a polynomial algorithm to test if a graph is Berge (or even whether testing Bergeness belongs to NP). We give an algorithm to answer this, with running time $O(|V(G)|^9)$.

A graph G is *perfect* if the chromatic number of H equals the size of the largest clique in H, for every induced subgraph H of G. Perfect graphs are of interest for many reasons, and in joint work with Neil Robertson and Robin Thomas, two of us showed in a recent paper [1] that a graph is Berge if and only if it is perfect. So we can test whether G is perfect by applying an algorithm to test whether G is Berge, and being able to test whether a graph is perfect might have practical applications, as well as being of theoretical interest. However, the algorithm we give here for testing Bergeness is independent of the theorem of [1].

Here is an outline of the algorithm. It makes use of "cleaning", a technique first used by Conforti and Rao [4] to recognize linear balanced matrices. (Cleaning is also a key step in the even hole recognition algorithm obtained jointly by two of us with Conforti and Kapoor [2]; and indeed, the two-step cleaning algorithm there was an ancestor of Routine 3 in this paper.) With input a graph G, we would like to decide either that G is not Berge, or that G contains no odd hole. (To test Bergeness, we just run this algorithm on G and then again on the complement of G.) If there is an odd hole in G, then there is a shortest one, say C. A vertex of the remainder of G is C-*major* if its set of neighbours in C is not a subset of the vertex set of any 3-vertex path of C; and C is *clean* (in G) if there are no C-major vertices in G. If there happens to be a clean shortest odd hole in G, then it stands out and can be detected relatively easily; and that essentially is the first step of our algorithm, a routine to test whether there is a clean shortest odd hole. The remainder of the algorithm consists of reducing the general problem to the "clean" case that was just handled. If C is a shortest odd hole in G, let us say a subset X of $V(G)$ is a *cleaner* for C if $X \cap V(C) = \emptyset$ and every C-major vertex belongs to X. Thus if X is a cleaner for C then C is a clean hole in $G \setminus X$. The idea of the remainder of the algorithm is to generate polynomially many subsets of $V(G)$, such that if there is a shortest odd hole C in G, then one of the subsets will be a cleaner for C. If we can do that, then we delete each of these subsets in turn, thereby generating polynomially many induced subgraphs; and we know that there is an odd hole in G if and only if in one of these subgraphs there is a clean shortest odd hole. Thus we can decide whether G has an odd hole by testing whether any of these subgraphs has a clean shortest odd hole.

In order to reduce the running time, it turns out to be advantageous not to do exactly what we just described, but to allow the subsets to meet the shortest odd hole within some 3-vertex path. Let C be a shortest odd hole in G. We say a subset X of $V(G)$ is a *near-cleaner* for C if X contains all C-major vertices, and $X \cap V(C)$ is a subset of the vertex set of some 3-vertex path of C. For us it is much faster to generate subsets such that one is guaranteed to be a near-cleaner, than if we really require one to be a cleaner; but we have to pay for it with added complications in the first step (the clean shortest odd hole detector), because now we have to detect a clean but "slightly-damaged" shortest odd hole. Nevertheless, the tradeoff is worth it.

How can we generate the polynomially many subsets such that one is a near-cleaner for a shortest odd hole C, without knowledge of C? This we only know how to do if C is "amenable", so let us define

1

that next. A subset $X \subseteq V(G)$ is *anticonnected* if the subgraph of $\overline{G}$ induced on X is connected. A vertex v is *X-complete* if $v \in V(G) \setminus X$ is adjacent to every vertex in X, and an edge is X-complete if both its ends are X-complete. A hole C in G is *amenable* if

- C is a shortest odd hole in G, of length at least 7, and

- for every anticonnected set X of C-major vertices, there is an X-complete edge in C.

We know how to clean if there is an amenable hole. Finally, there is the possibility that the input graph has an odd hole but no amenable hole. To handle this we first run some tests which in this case will detect directly that G is not Berge.

Thus, the algorithm falls naturally into three parts (numbered in the order in which they were discovered, and in the order in which it is most convenient to explain them, not the order in which they are applied). The first is the clean shortest odd hole detector, modified to allow for near-cleaners. More precisely:

Routine 1: A polynomial algorithm with input a graph G and a subset X of $V(G)$, such that if X is a near-cleaner for a shortest odd hole in G, then the algorithm will discover an odd hole in G.

The second part is the amenability test:

Routine 2: A polynomial algorithm such that, if some shortest odd hole in G is not amenable, the algorithm will discover that G is not Berge.

Finally, we need the cleaning process:

Routine 3: A polynomial algorithm that outputs polynomially many subsets of $V(G)$, such that if C is an amenable hole in G, then one of the subsets is a near-cleaner for C.

From these parts, we construct an algorithm to test Bergeness as follows. First we run Routine 2; and if we do not detect that G is not Berge, from now on we know that every shortest odd hole is amenable. Now we run Routine 3, and get the polynomially many subsets. For each of them (say X) in turn, we run Routine 1 on the pair G, X. If we still have not decided that G is not Berge, we repeat everything on the complement graph; and if there too we cannot deduce that G is not Berge, then it is Berge and we output that.

Incidentally, there is another interesting open question in this area — is there a polynomial algorithm to test if G contains an odd hole? The algorithms of Routines 1 and 3 work equally well for that question; they take no notice of odd holes in $\overline{G}$. However, there are places in the algorithm for Routine 2 where we stumble over an odd hole in $\overline{G}$ and stop, declaring G non-Berge, and we are currently unable to eliminate that feature; so the question of testing just for odd holes in G remains open.

This paper is the product of the research of two groups that were working separately on the problem, Cornuéjols, Liu and Vušković (CLV), and Chudnovsky and Seymour (ChS). The approach of both groups was based on cleaning a shortest odd hole, and there was a great deal of overlap in their results. Both groups simultaneously obtained Routine 1, in quite different ways; ChS obtained Routines 2 and 3; and then CLV obtained Routine 3, with a much better method. What is presented here is the work of ChS on Routines 1 and 2, and the work of CLV on Routine 3. In the appendix we also give the version of Routine 1 due to CLV, which may be of some independent interest.

2 Testing for pyramids

In this section we begin on the algorithm for Routine 1. Let us be more precise. All graphs in this paper are finite, and simple. The vertex- and edge-sets of a graph G are denoted by $V(G), E(G)$. A subset $X \subseteq V(G)$ is *connected* if the subgraph of G induced on X is connected. If $X \subseteq V(G)$, a *component* of X means a maximal nonnull connected subset of X. A *path* in G is an induced subgraph that is connected, with at least one vertex, no cycle, and no vertex of degree > 2. The *ends* of a path are defined as usual. We denote the set of internal vertices of a path P (that is, its *interior*) by P^*. The *length* of a path or hole is the number of edges in it, and a path or hole is *odd* if it has odd length and *even* otherwise. If $u, v \in V(G)$, in the same component, we denote the length of the shortest path in G between them by $d_G(u, v)$.

A *triangle* in a graph G means a set of three pairwise adjacent vertices of G. A *pyramid* in G is an induced subgraph formed by the union of a triangle $\{b_1, b_2, b_3\}$, a fourth vertex a, and three paths P_1, P_2, P_3, satisfying:

- for $i = 1, 2, 3$, P_i is between a and b_i

- for $1 \leq i < j \leq 3$, a is the only vertex in both P_i, P_j, and $b_i b_j$ is the only edge of G between $V(P_i) \setminus \{a\}$ and $V(P_j) \setminus \{a\}$

- a is adjacent to at most one of b_1, b_2, b_3.

So every pyramid is a subdivision of K_4. Note that the pyramid is determined by a knowledge of P_1, P_2, P_3, and we call it the pyramid *formed* by P_1, P_2, P_3. If there exist a, b_1, b_2, b_3 etc as above, we say that a can be *linked onto* the triangle $\{b_1, b_2, b_3\}$, *via* the paths P_1, P_2, P_3. It is easy to see that any graph containing a pyramid contains an odd hole. Our objective in this section is to give a polynomial algorithm to test whether G contains a pyramid.

If K is a pyramid, formed by three paths P_1, P_2, P_3 linking a onto b_1, b_2, b_3 respectively, we say its *frame* is the 10-tuple

$$a, b_1, b_2, b_3, s_1, s_2, s_3, m_1, m_2, m_3,$$

where

- for $i = 1, 2, 3$, s_i is the neighbour of a in P_i

- for $i = 1, 2, 3$, $m_i \in V(P_i)$ satisfies $d_{P_i}(a, m_i) - d_{P_i}(m_i, b_i) \in \{0, 1\}$.

A pyramid K in G is *optimal* if there is no pyramid K' with $|V(K')| < |V(K)|$. If there is a pyramid in G, then there is an optimal one, and optimal pyramids have some special structure that helps us detect them.

2.1 *Let K be an optimal pyramid, with frame $a, b_1, b_2, b_3, s_1, s_2, s_3, m_1, m_2, m_3$. Let S_1, T_1 be the subpaths of P_1 from m_1 to s_1, b_1 respectively. Let F be the set of all vertices nonadjacent to each of s_2, s_3, b_2, b_3.*

1. *Let Q be a path between s_1 and m_1 with interior in F, and with minimum length over all such paths. Then a-s_1-Q-m_1-T_1-b_1 is a path (say P_1'), and P_1', P_2, P_3 form an optimal pyramid.*

3

2. *Let Q be a path between m_1 and b_1 with interior in F, and with minimum length over all such paths. Then a-s_1-S_1-m_1-Q-b_1 is a path (say P'_1), and P'_1, P_2, P_3 form an optimal pyramid.*

Analogous statements holds for P_2, P_3.

Proof. Note first that from the choice of m_1, we have

$$|E(S_1)| \leq |E(T_1)| \leq |E(S_1)| + 1.$$

Let U be the path induced on $V(P_2 \cup P_3) \setminus \{b_2, b_3\}$. Let us prove the first statement. If $s_1 = m_1$ or s_1, m_1 are adjacent, then the claim holds trivially, so we assume that s_1, m_1 are distinct and nonadjacent. Hence S_1 has length ≥ 2, and therefore T_1 has length ≥ 2, and in particular $m_1 \neq b_1$. From the choice of Q, it follows that $|E(Q)| \leq |E(S_1)|$. Let Q have vertices q_1-$\cdots$-q_n, where $q_1 = s_1$ and $q_n = m_1$. Then $n \geq 3$.

Suppose first that none of $q_2, \ldots, q_{n-1}$ belong to or have neighbours in U. Then there is a path P'_1 between a, b_1 with interior in $V(Q \cup T_1)$, of length at most $1 + |E(Q)| + |E(T_1)|$. Hence P'_1, P_2, P_3 form a pyramid, and from the optimality of K it follows that P'_1 has length at least that of P_1. So

$$|E(P'_1)| \geq 1 + |E(S_1)| + |E(T_1)| \geq 1 + |E(Q)| + |E(T_1)| \geq |E(P'_1)|$$

and therefore equality holds throughout, and in particular, a-s_1-Q-m_1-T_1-b_1 is the path P'_1, and the pyramid formed by P'_1, P_2, P_3 is optimal. Thus in this case the claim holds.

We may therefore assume that for some k with $2 \leq k \leq n-1$, q_k belongs to or has neighbours in U. Choose such a value of k, maximum. We claim that none of $q_k, \ldots, q_n$ belongs to U; for if $k < n-1$ this follows from the maximality of k, and if $k = n-1$ it follows since $q_n = m_1$ has no neighbour in U. From the choice of Q, none of $q_k, \ldots, q_{n-1}$ is adjacent to any of b_2, b_3.

Suppose that q_k has nonadjacent neighbours in U. Then q_k can be linked onto $\{b_1, b_2, b_3\}$ via a path from q_k to b_1 with interior in $\{q_{k+1}, \ldots, q_n\} \cup V(T_1)$, and two paths with interior in $V(U)$ from q_k to b_2, b_3 respectively. Since the first path is strictly shorter than P_1 (since $k \geq 2$ and $|E(Q)| \leq |E(S_1)|$), and the sum of the lengths of the other two is at most the sum of the lengths of P_1, P_2, this contradicts the optimality of K.

Next suppose that q_k has a unique neighbour in U, say x. Then x can be linked onto $\{b_1, b_2, b_3\}$, via a path from x to b_1 with interior in $\{q_k, \ldots, q_n\} \cup V(T_1)$, and two paths with interior in $V(U)$, from x to b_2, b_3 respectively, contrary to the optimality of K (since $k \geq 2$ and $|E(Q)| \leq |E(S_1)|$).

So q_k has exactly two neighbours in U, and they are adjacent. Since q_k is nonadjacent to s_2, s_3, it follows that q_k is nonadjacent to a, and we may assume that q_k has two adjacent neighbours in P_2, different from a, s_2, b_2. Let X be the subpath of P_2 between a and the neighbour of q_k that is closer to a in P_2 (say x), and let Y be the subpath of P_2 between b_2 and the neighbour of q_k that is closer to b_2 (say y). Then a can be linked onto $\{q_k, x, y\}$, via a path from a to q_k with interior in $V(S_1) \cup \{q_{k+1}, \ldots, q_n\}$, a-X-x and a-P_3-b_3-b_2-Y-y. Since the sum of the lengths of the second and third paths equals the sum of the lengths of P_2, P_3, and the first path has length

$$\leq 1 + |E(S_1)| + n - k \leq 2|E(S_1)| \leq |E(S_1)| + |E(T_1)| < |E(P_1)|,$$

this contradicts the optimality of K. This proves the first statement of 2.1.

Now we prove the second statement. If $m_1 = b_1$ or m_1, b_1 are adjacent, the claim holds trivially, so we assume m_1, b_1 are distinct and nonadjacent. Hence T_1 has length ≥ 2, and therefore S_1 has

4

length ≥ 1. From the choice of Q, $|E(Q)| \leq |E(T_1)|$. Let Q have vertices $q_1\text{-}\cdots\text{-}q_n$, where $q_1 = m_1$ and $q_n = b_1$; then $n \geq 3$, since m_1, b_1 are distinct and nonadjacent. Suppose that none of $q_2, \ldots, q_{n-1}$ belong to or have neighbours in U. Then from the optimality of K, it follows that P_1 is a shortest path between a, b_1 with interior in $V(P_1) \cup \{q_2, \ldots, q_{n-1}\}$; and so Q, T_1 have the same length, and there are no edges between $\{q_2, \ldots, q_{n-1}\}$ and $V(S_1 \setminus m_1)$, and therefore the claim holds.

So we may assume that q_k belongs to or has a neighbour in U, for some k with $2 \leq k \leq n-1$. Choose k minimum. It follows that none of $q_1, \ldots, q_k$ belong to U. Let R be a path from q_k to b_1 with interior in $\{q_1, \ldots, q_k\} \cup V(T_1)$. We claim that $|E(R)| \leq |E(P_1)| - 2$. This is clear if q_k is adjacent to b_1, since P_1 has length ≥ 4. If q_k is not adjacent to b_1 then

$$k - 1 \leq n - 3 \leq |E(T_1)| - 2 \leq |E(S_1)| - 1,$$

and so

$$|E(R)| \leq k - 1 + |E(T_1)| \leq |E(S_1)| + |E(T_1)| - 1 = |E(P_1)| - 2.$$

In either case, $|E(R)| \leq |E(P_1)| - 2$ as claimed.

From the choice of Q, q_k is nonadjacent to b_2, b_3. Suppose first that q_k has nonadjacent neighbours in U. Then q_k can be linked onto $\{b_1, b_2, b_3\}$ via R and two paths with interior in U from q_k to b_2, b_3 respectively. Since R is strictly shorter than P_1, and the sum of the lengths of the other two is at most the sum of the lengths of P_2, P_3, this contradicts the optimality of K.

Next suppose that q_k has a unique neighbour in U, say x. Then x can be linked onto $\{b_1, b_2, b_3\}$, via $x\text{-}q_k\text{-}R\text{-}b_1$ and two paths with interior in U from x to b_2, b_3 respectively, again contrary to the optimality of K (since $|E(R)| \leq |E(P_1)| - 2$).

This q_k has exactly two neighbours in U, and they are adjacent. Since q_k is nonadjacent to s_2, s_3, follows that q_k is nonadjacent to a, and we may assume that q_k has two adjacent neighbours in P_2, different from a, s_2, b_2. Let X be the subpath of P_2 between a and the neighbour of q_k that is closer to a in P_2 (say x), and let Y be the subpath of P_2 between b_2 and the neighbour of q_k that is closer to b_2 (say y). Then a can be linked onto $\{q_k, x, y\}$, via a path from a to q_k with interior in $\{q_1, \ldots, q_{k-1}\} \cup V(S_1)$, $a\text{-}X\text{-}x$ and $a\text{-}P_3\text{-}b_3\text{-}b_2\text{-}Y\text{-}y$. Since the first path is strictly shorter than P_1, and the sum of the lengths of the second two paths equals the sum of the lengths of P_2, P_3, again this contradicts the optimality of K. This proves the second statement of 2.1. ∎

We use the previous lemma to prove the following.

2.2 *There is an algorithm with the following specifications:*

Input: *A graph G.*

Output: *Either it finds a pyramid (and hence an odd hole) in G, or it determines that G contains no pyramid.*

Running time: $O(|V(G)|^9)$.

Proof. Here is an algorithm. Enumerate all 6-tuples $b_1, b_2, b_3, s_1, s_2, s_3$ that satisfy the following conditions:

- for $1 \leq i < j \leq 3$, $\{b_i, s_i\}$ is disjoint from $\{b_j, s_j\}$, and $b_i b_j$ is the unique edge between them

- there is a vertex a adjacent to all of s_1, s_2, s_3 and to at most one of b_1, b_2, b_3, such that for $1 \leq i \leq 3$, if a is adjacent to b_i then $s_i = b_i$ (let us call such a vertex a an *apex* for the 6-tuple).

(We can find all such 6-tuples in time $O(|V(G)|^7)$.)

For each such choice of $b_1, b_2, b_3, s_1, s_2, s_3$ we do the following. Let $M = V(G)\backslash\{b_1, b_2, b_3, s_1, s_2, s_3\}$. For each $m \in M$, find a shortest path $S_1(m)$ between s_1, m such that s_2, s_3, b_2, b_3 have no neighbours in its interior, if such a path exists. Find a shortest path $T_1(m)$ between m, b_1 such that s_2, s_3, b_2, b_3 have no neighbours in its interior, if such a path exists. Find $S_2(m), T_2(m), S_3(m), T_3(m)$ similarly. (To find these paths, for a given 6-tuple $b_1, b_2, b_3, s_1, s_2, s_3$, but for all m, takes time $O(|V(G)|^2)$.)

Next, for each $m \in M \cup \{b_1\}$ let $P_1(m)$ be defined as follows. If $s_1 = b_1$ let $P_1(b_1)$ be the one-vertex path with vertex b_1, and let $P_1(m)$ be undefined for each $m \in M$. Now assume that $s_1 \neq b_1$. Then $P_1(b_1)$ is undefined; and for each $m \in M$, test whether all the following are true:

- m is nonadjacent to b_2, b_3, s_2, s_3

- $S_1(m), T_1(m)$ both exist

- $V(S_1(m) \cap T_1(m)) = \{m\}$

- there are no edges between $V(S_1(m) \setminus m)$ and $V(T_1(m) \setminus m)$.

If so, then s_1-$S_1(m)$-m-$T_1(m)$-b_1 is a path; call it $P_1(m)$ (and otherwise $P_1(m)$ is undefined). Define $P_2(m), P_3(m)$ similarly. (Finding $P_1(m), P_2(m), P_3(m)$ for all m takes time $O(|V(G)|^3)$.)

Our goal is now to check for each triple m_1, m_2, m_3 whether the three paths $P_i(m_i)$ $(i = 1, 2, 3)$ form a pyramid, for some suitable choice of a vertex a. But with care we can significantly reduce the running time, so let us do it carefully.

If $1 \leq i < j \leq 3$, we say that (m_i, m_j) is a *good* (i, j)-pair if $m_i \in M \cup \{b_i\}$, $m_j \in M \cup \{b_j\}$, $P_i(m_i), P_j(m_j)$ both exist, and the sets $V(P_i(m_i)), V(P_j(m_j))$ are disjoint and $b_i b_j$ is the only edge between them. Next we want to find the list of all good $(1, 2)$-pairs. (And we need to do it in time $O(|V(G)|^3)$, so the obvious method is not fast enough.) For each $m_1 \in M \cup \{b_1\}$, we find the set of all m_2 such that (m_1, m_2) is good, in two stages as follows. If $P_1(m_1)$ does not exist, there are no such good pairs. If it exists, colour black the vertices of M that either belong to $P_1(m_1)$ or have a neighbour in $P_1(m_1)$, and colour all other vertices white. (This takes time $O(|V(G)|^2)$.) Then for each $m_2 \in M \cup \{b_2\}$, test whether $P_2(m_2)$ exists and contains no black vertices. (For each m_2 this takes linear time, so doing it for all m_2 takes time $O(|V(G)|^2)$.) Repeating this for all m_1, we compute the set of all good $(1, 2)$-pairs (in time $O(|V(G)|^3)$). Repeat to find the good $(1, 3)$-pairs and $(2, 3)$-pairs. Now we examine all triples m_1, m_2, m_3 such that $m_i \in M_i \cup \{b_i\}$ for $i = 1, 2, 3$ and test whether (m_i, m_j) is a good (i, j)-pair for $1 \leq i < j \leq 3$. (For each triple this takes constant time, so altogether it again takes time $O(|V(G)|^3)$.) If we find a triple such that all three pairs are good, we output that G contains a pyramid and stop.

After examining all choices of $b_1, b_2, b_3, s_1, s_2, s_3$, output that G contains no pyramid. (Each choice of $b_1, b_2, b_3, s_1, s_2, s_3$ takes total time $O(|V(G)|^3)$ to process and since there are $O(|V(G)|^6)$ such choices, the total running time is $O(|V(G)|^9)$.)

Now we need to prove that the algorithm works correctly. Suppose first that it outputs that G contains a pyramid. Therefore for some choice of $b_1, b_2, b_3, s_1, s_2, s_3$, we know the following:

- for $1 \leq i < j \leq 3$, $\{b_i, s_i\}$ is disjoint from $\{b_j, s_j\}$, and $b_i b_j$ is the unique edge between them, and in particular, $\{b_1, b_2, b_3\}$ is a triangle

6

- there is a vertex a adjacent to all of s_1, s_2, s_3 and to at most one of b_1, b_2, b_3, such that for $1 \le i \le 3$, if a is adjacent to b_i then $s_i = b_i$,

- with notation as before, for $i = 1, 2, 3$ there exists $m_i \in M \cup \{b_i\}$, such that (m_i, m_j) is a good (i, j)-pair for $1 \le i < j \le 3$.

Then for $i = 1, 2, 3$, $P_i(m_i)$ is a path between s_i and b_i, these three paths are vertex-disjoint, and the only edges between them join two of $\{b_1, b_2, b_3\}$. Moreover there is a vertex a, adjacent to at most one of b_1, b_2, b_3, and with a neighbour in each $P_i(m_i)$. Hence a can be linked onto the triangle $\{b_1, b_2, b_3\}$, via paths with interior in $V(P_i(m_i))(i = 1, 2, 3)$, and therefore G contains a pyramid. So in this case the output of the algorithm was correct.

For the converse, suppose that G in fact does contain a pyramid, and let an optimal pyramid in G be formed by P_1, P_2, P_3, with frame

$$a, b_1, b_2, b_3, s_1, s_2, s_3, m_1, m_2, m_3.$$

We can assume that the algorithm inspects the 6-tuple $b_1, b_2, b_3, s_1, s_2, s_3$, for if not then it has already detected some pyramid and stopped, and the output is correct. Let us examine what the algorithm does when it inspects this 6-tuple. Since a is an apex for this 6-tuple, the algorithm will proceed to search for paths $S_i(m_i), T_i(m_i)$ for $1 \le i \le 3$, and since such paths exist, it will find them. By six applications of 2.1, it follows that the union of these six paths (together with the vertex a, the edges as_i, and the edges $b_i b_j$) is an optimal pyramid; and so the algorithm will detect this pyramid, and output correctly that G contains a pyramid. This proves 2.2. ∎

Since this is the slowest part (or one of them) of the algorithm to test Bergeness, it would be nice to make it faster, but at the moment we don't see how. Here are three encouraging observations, each of which looks at first sight as if it brings the running time down to $O(|V(G)|^8)$:

- Let P_1, P_2, P_3 be the optimal pyramid, and let P_1 be the shortest of the three paths, with second vertex s_1. Then any minimum length path between s_1 and b_1 containing no neighbours of b_2, b_3 can be used in place of $P_1 \setminus s_1$; in other words we don't need to "guess" the middle vertex m_1 before we can find the path.

- For any one of the three paths, say P_3, if we have figured out P_1, P_2 correctly, we don't need the best possible P_3; to be sure that G contains a pyramid, it is enough to detect any path from a to b_3 that does not pass through vertices with neighbours in P_1, P_2.

- We don't really need to find a pyramid; it is enough to find an odd hole. And if there is an optimal pyramid, there is an odd hole formed by the union of two of the paths $P_i(m_i)$ $(i = 1, 2, 3)$. So it is enough to examine each possible pair of paths, rather than trying to examine triples of paths.

But none of these is enough, as far as we can see; the running time remains $O(|V(G)|^9)$.

3 Finding jewels

Before we explain Routine 1, there is another configuration we need to eliminate. We say a sequence $v_1, \ldots, v_5, P$ is a *jewel* in G if $v_1, \ldots, v_5$ are distinct vertices, $v_1 v_2, v_2 v_3, v_3 v_4, v_4 v_5, v_5 v_1$ are

7

edges, v_1v_3, v_2v_4, v_1v_4 are nonedges, and P is a path of G between v_1, v_4 such that v_2, v_3, v_5 have no neighbours in P^*.

3.1 *There is an algorithm with the following specifications:*

Input: *A graph G.*

Output: *Decides whether there is a jewel in G.*

Running time: $O(|V(G)|^6)$.

Proof. The obvious implementation (enumerate all choices of $v_1, \ldots, v_5$ and check them) has running time $O(|V(G)|^7)$, but we can gain a little bit as follows. Enumerate all 3-tuples v_2, v_3, v_5 of distinct vertices such that v_2v_3 is an edge. For each choice of v_2, v_3, v_5, find the set F of all vertices nonadjacent to each of v_2, v_3, v_5, and find all its components. Find the set X_1 of all vertices adjacent to v_2, v_5 and not to v_3, and for each $v_1 \in X_1$ and each component of F' of F, record whether v_1 has a neighbour in F'. Do the same for the set X_2 of all vertices adjacent to v_3, v_5 and not to v_2. Then test if there exist $v_1 \in X_1$, $v_4 \in X_2$, and a component F' of F, such that v_1, v_4 are nonadjacent and both have neighbours in F'. If so then output that G contains a jewel. If after examining all choices we still have not found a jewel, then none exists; output that fact. This proves 3.1. ∎

We observe also that:

3.2 *If there is a jewel in G, then there is an odd hole in G.*

Proof. Let $v_1, \ldots, v_5, P$ be a jewel. If P is odd, then v_1-P-v_4-v_5-v_1 is an odd hole, and otherwise v_1-P-v_4-v_3-v_2-v_1 is an odd hole. This proves 3.2. ∎

4 The clean shortest odd hole detector

In this section we prove a theorem used to show the correctness of our algorithm for Routine 1 (which is completed in the next section). Let C be a shortest odd hole in G. We recall that, if $v \in V(G) \setminus V(C)$, we say v is C-*major* if the set of its neighbours in C is not contained in any 3-vertex path of C; and C is *clean* if no vertex is C-major.

4.1 *Let G be a graph containing no jewel or pyramid, and let C be a clean shortest odd hole in G. Let $u, v \in V(C)$ be distinct and nonadjacent, and let L_1, L_2 be the two subpaths of C joining u, v, where $|E(L_1)| < |E(L_2)|$. Then:*

- *L_1 is a shortest path in G between u, v, and*

- *for any shortest path P in G between u, v, $P \cup L_2$ is a shortest odd hole in G, and it is clean.*

Proof. Assume that the theorem is false, and choose a shortest odd hole C and vertices $p_1, \ldots, p_k$, with k as small as possible such that the following holds: there exist $u, v \in V(C)$ such that u-p_1-$\cdots$-p_k-v is a path P of G, and with L_1, L_2 defined as in the theorem, either L_1 has length $> k+1$, or it has length $k+1$ and $P \cup L_2$ is not a clean shortest odd hole. (We refer to this property

8

as the "minimality of k".) For fixed C and $p_1, \ldots, p_k$ choose u, v in addition such that $|E(L_2)|$ is as small as possible. Evidently P is a shortest path between u, v.

Assign C an orientation, clockwise say, and for any two distinct vertices x, y in C, let $C(x, y)$ be the clockwise path in C from x to y, when it exists (that is, unless y immediately precedes x in the clockwise order). We may assume that $L_1 = C(u, v)$. Let C have vertices c_1-$\cdots$-c_{2n+1} in clockwise order, where $c_1 = u$ and $c_m = v$; and therefore $m \le n + 1$. From the hypothesis,

$$k + 1 = d_G(u, v) \le d_C(u, v) = m - 1 \le n.$$

(We recall that $d_G(u, v)$ is the length of a shortest path in G between u, v.)

(1) $k \ge 2$, and consequently $m \ge 4$ and $n \ge 3$.

For assume that $k = 1$. Since p_1 is not C-major, it follows that $m = 3$, and the only neighbours of p_1 in C are c_1, c_3 and possibly c_2. In particular, c_1-p_1-c_3-$\cdots$-c_{2n+1}-c_1 is a hole C'. Since it has the same length as C, we deduce that C' is a shortest odd hole. Since the theorem is not satisfied, C' is not clean, and so there is a vertex w that is C'-major. Since w is not C-major, it follows that $C' \ne C$, and so $p_1 \notin V(C)$; and for the same reason, w is adjacent to p_1. The neighbours of w in C do not all lie in the path c_2-c_3-c_4, since w is C'-major; and similarly they do not all lie in c_{2n+1}-c_1-c_2, and not all in c_1-c_2-c_3. Since they do all lie in some 3-vertex path of C, we may assume from the symmetry that w is nonadjacent to both c_1, c_2. Choose i, j with $3 \le i \le j \le 2n+1$, minimum and maximum respectively such that w is adjacent to c_i, c_j. Hence $j \le i + 2$, and $j \ge 5$. The hole w-c_j-$\cdots$-c_{2n+1}-c_1-p_1-w is shorter than C and therefore even, and so j is odd. If $i > 3$ then similarly i is even, and since $j - i \le 2$ it follows that $j = i + 1$; but then the paths p_1-w, p_1-c_3-$\cdots$-c_i, p_1-c_1-c_{2n+1}-$\cdots$-c_{i+1} form a pyramid in G, a contradiction. So $i = 3$, and therefore $j = 5$. But then c_1, c_2, c_3, w, p_1 and the path w-c_5-$\cdots$-c_{2n+1}-c_1 form a jewel, a contradiction. Thus $k \ge 2$. Since $k + 1 \le m - 1 \le n$, this proves (1).

(2) The sets $P^*, C(v, u)^*$ are disjoint, and there are no edges between $\{p_2, \ldots, p_{k-1}\}$ and $C(v, u)^*$.

For suppose not. Then for some j with $m + 1 \le j \le 2n + 1$, there exist paths P_1, P_2 from c_j to u, v respectively, with interior in P^*, both strictly shorter than P. Suppose first that $j = 2n + 1$. Then

$$d_C(c_{2n+1}, v) = \min(m, 2n + 1 - m) \ge m - 1 \ge |E(P)| > |E(P_2)|,$$

and since P_2 is strictly shorter than P, this contradicts the minimality of k. Thus $j \le 2n$ and similarly $j \ge m + 2$. In particular, P_1, P_2 both have length at least two. Now

$$|E(P_1)| + |E(P_2)| \le k + 3 \le m + 1 \le 2n + 3 - m < (2n + 3 - j) + (j + 1 - m),$$

and so either P_1 has length at most that of $C(c_j, u)$, or P_2 has length at most that of $C(v, c_j)$. From the symmetry between u, v we may assume the first. From the minimality of k it follows that

$$d_G(c_j, u) = d_C(c_j, u) = \min(2n + 2 - j, j - 1).$$

But

$$d_G(c_j, u) \le |E(P_1)| < |E(P)| \le m - 1 \le j - 1,$$

9

so $d_G(c_j, u) = 2n + 2 - j$. Since P_1 has length at most $2n + 2 - j$, it follows that P_1 is a shortest path in G between c_j, u. Let u' be the neighbour of c_j in P_1. The minimality of k implies that u-$C(u, c_j)$-c_j-P_1-u is a clean shortest odd hole (C' say) in G, and in particular, u', v are nonadjacent. Orient C' such that the orientations of C, C' agree on the common subpath $C(u, v)$. There is a subpath P' of P between u', v, of length $k + 2 - |E(P_1)| = k + j - 2n$. Since P' is strictly shorter than P (because c_j, u are nonadjacent), it follows from the minimality of k (applied to C', P') that one of the paths $C'(u', v), C'(v, u')$ has length at most $k + j - 2n$, and in particular is strictly shorter than P. But $C'(u', v)$ includes $C(u, v)$ and therefore is not strictly shorter than P; and $C'(v, u')$ has length

$$j - m + 1 \geq j - m + 1 - 2(n - m + 1) - (m - k - 2) > k + j - 2n,$$

a contradiction. This proves (2).

(3) *Either c_1 is the only neighbour of p_1 in C, or c_1, c_2 are the only neighbours of p_1 in C, or $m = n + 1$ and c_1, c_{2n+1} are the only neighbours of p_1 in C; and in particular, $p_1 \notin V(C)$. The analogous statement holds for p_k.*

For suppose first that p_1 has two nonadjacent neighbours $x, z \in V(C)$. Since p_1 is not C-major, we may assume that $C(x, z)$ has length 2 and contains all neighbours of p_1 in C (and, if $p_1 \in V(C)$, it is the middle vertex of $C(x, z)$). Let y be the middle vertex of $C(x, z)$; then $u \in \{x, y, z\}$, and since u, v are nonadjacent and p_1, v are nonadjacent by (1), it follows that $v \neq x, y, z$. Now p_1-z-$C(z, x)$-x-p_1 is a hole C' of the same length as C, and hence is a shortest odd hole, and it is clean, by (1) and the minimality of k. Since $d_G(p_1, v) = k$, it follows from the minimality of k that $d_{C'}(p_1, v) = k$. Since $v \neq y$, it follows that $d_{C'}(p_1, v) = d_C(y, v)$, and therefore $d_C(y, v) = k$. Since $C(u, v)$ has length $\leq n$, and contains z, it follows that $d_C(z, v) \leq d_C(y, v) = k$. But $d_C(u, v) \geq k + 1$, and $u \in \{x, y, z\}$; and therefore $u = x$. From the minimality of k (applied to p_1-$\cdots$-p_k-v and C'), it follows that u-p_1-$\cdots$-p_k-v-$C(v, u)$-u is a clean shortest odd hole. But then the theorem holds, a contradiction. Hence p_1 does not have two nonadjacent neighbours in C, and in particular, $p_1 \notin V(C)$. Since p_1 is adjacent to c_1, we may assume it is also adjacent to c_{2n+1}, for otherwise the claim holds. Suppose that $m \leq n$. Since c_{2n+1} is nonadjacent to $p_2, \ldots, p_{k-1}$ by (2), and nonadjacent to p_k since p_k does not have two nonadjacent neighbours in C, it follows that c_{2n+1}-p_1-$\cdots$-p_k-c_m is a path; and by the second minimization in the choice of u, v (minimizing $|E(L_2)|$) applied to this path, it follows that

$$d_C(c_{2n+1}, c_m) = d_G(c_{2n+1}, c_m) \leq k + 1.$$

But $d_C(c_{2n+1}, c_m) = m \geq k + 2$, a contradiction. Hence $m = n + 1$. This proves (3).

(4) *There are no edges between P^* and $C(v, u)^*$.*

For suppose there are edges between P^* and $C(v, u)^*$. From (2) and (3), we may assume that p_1 is adjacent to c_{2n+1} and $m = n + 1$. Let P' be the path c_{2n+1}-p_1-$\cdots$-p_k-v. If $d_G(c_{2n+1}, v) < k + 1$, then from the minimality of k, $d_C(c_{2n+1}, v) < k + 1 \leq n$, a contradiction since $m = n + 1$. So $d_G(c_{2n+1}, v) \geq k + 1$, and therefore P' is a shortest path in G between c_{2n+1}, v. Hence there is symmetry between c_1, c_{2n+1}, and from (2) applied under this symmetry we deduce that P^* is disjoint from $C(u, v)^*$, and there are no edges between $\{p_2, \ldots, p_{k-1}\}$ and $C(u, v)^*$. Consequently there are no edges between P^* and C except for $p_1 c_1, p_1 c_{2n+1}$ and possibly edges incident with p_k. Since G

10

contains no pyramid, c_m cannot be linked onto the triangle $\{p_1, c_1, c_{2n+1}\}$, and therefore p_k has at least two neighbours in C; by (3), p_k has exactly two neighbours in C and they are adjacent; and from the symmetry between c_1, c_{2n+1} we may assume that the second neighbour of p_k is c_{m-1}. If $k + m$ is even then c_1-$C(c_1, c_{m-1})$-p_k-$\cdots$-p_1-c_1 is an odd hole of length $k + m - 1 < 2n + 1$, while if $k + m$ is odd then c_m-$C(c_m, c_{2n+1})$-c_{2n+1}-p_1-$\cdots$-p_k-c_m is an odd hole of length $2n - m + k + 2 < 2n + 1$, in both cases a contradiction. This proves (4).

(5) $k + 2 < m$.

For suppose not. Certainly $k + 2 \leq m$, and so $P, C(u, v)$ have the same length. By (4),

$$c_1\text{-}p_1\text{-}\cdots\text{-}p_k\text{-}c_m\text{-}c_{m+1}\text{-}\cdots\text{-}c_{2n+1}\text{-}c_1$$

is a hole, C' say; and it is a shortest odd hole, since $P, C(u, v)$ have the same length. Consequently it is not clean, since the theorem is not satisfied; let $w \in V(G)$ be C'-major. Since it is not C-major, it is adjacent to at least one of $p_1, \ldots, p_k$. Since P is a shortest path between u, v, the neighbours of w in P lie in a 3-vertex subpath of P; and since w is C'-major, it follows that w is adjacent to at least one of $c_{m+1}, \ldots, c_{2n+1}$.

Suppose that w has two nonadjacent neighbours in the path $C(v, u)$. Since w is not C-major, we may assume that the neighbours of w in C are c_i, c_{i+2}, and possibly c_{i+1} where $m \leq i \leq 2n - 1$. By (1) and the minimality of k, the hole obtained from C by replacing c_{i+1} by w is clean; but there are edges between w and $p_1, \ldots, p_k$, contradicting (4) applied to this hole. So w does not have two nonadjacent neighbours in $C(v, u)$.

Suppose that w is adjacent to none of $c_{m+2}, \ldots, c_{2n}$. Then we may assume w is adjacent to c_{2n+1}. Since it is not C-major, it has no neighbours in C except c_{2n+1} and possibly c_1, c_2. Choose i with $1 \leq i \leq k$ maximum such that w is adjacent to p_i. Since w is C'-major it follows that $i > 1$; and since $c_{2n+1}, c_1, p_1, p_2, w$ and the path p_2-$\cdots$-p_k-c_m-$\cdots$-c_{2n+1} do not form a jewel, it follows that $i > 2$. Hence $d_G(c_m, c_{2n+1}) \leq k$, and so from the minimality of k, $d_G(c_m, c_{2n+1}) = d_C(c_m, c_{2n+1})$. But

$$d_C(c_m, c_{2n+1}) = \min(m, 2n + 1 - m) > k,$$

a contradiction. So w is adjacent to one of $c_{m+2}, \ldots, c_{2n}$. Since it is not C-major, it is nonadjacent to all of $c_2, \ldots, c_{m-1}$. Since it does not have two nonadjacent neighbours in $C(v, u)$, it is nonadjacent to both c_1, c_m.

Choose i, j with $1 \leq i \leq j \leq k$, minimum and maximum respectively such that w is adjacent to p_i, p_j. Choose s, t with $m + 1 \leq s \leq t \leq 2n + 1$, minimum and maximum respectively such that w is adjacent to c_s, c_t. Since P is a shortest path between u, v it follows that $j - i \leq 2$. Since w does not have two nonadjacent neighbours in $C(v, u)$, it follows that $t - s \leq 1$. Since w is adjacent to one of $c_{m+2}, \ldots, c_{2n}$, it follows that $s \leq 2n$ and $t \geq m + 2$. Since

$$w\text{-}c_t\text{-}\cdots\text{-}c_{2n+1}\text{-}c_1\text{-}p_1\text{-}\cdots\text{-}p_i\text{-}w$$

is a hole of length $\leq 2n$, it is even, and so $t + i$ is even; and similarly $s - m + k - j$ is odd. Since $k = m - 2$ it follows that $s + j$ is odd. Consequently $(t - s) + (j - i)$ is odd.

Suppose that $i = j$. Then $t - s = 1$ since $t - s$ is odd; but then p_i can be linked onto the triangle $\{w, c_s, c_t\}$ via p_i-w and two subpaths of C', contradicting that G contains no pyramid. So $j > i$.

Similarly if $s = t$ then $j - i = 1$, and c_s can be linked onto $\{w, p_i, p_j\}$, again a contradiction. So $t > s$. Since $t - s \leq 1$ it follows that $t = s + 1$ and therefore $j = i + 2$, since $(t - s) + (j - i)$ is odd. But then the path $u\text{-}p_1\text{-}\cdots\text{-}p_i\text{-}w\text{-}p_j\text{-}\cdots\text{-}p_k\text{-}v$ is a path between u, v with the same length as P, and yet there are edges between its interior and $C(v, u)^*$, contrary to (4) applied to this path. This proves (5).

(6) *The sets $P^*, C(u, v)^*$ are disjoint, and there are no edges between $\{p_2, \ldots, p_{k-1}\}$ and $C(u, v)$.*

The argument is almost identical with that for (2). Suppose the claim is false; then there exists j with $2 \leq j \leq m - 1$, and paths P_1, P_2 from c_j to u, v respectively, with interior in P^*, and both strictly shorter than P. Since

$$|E(P_1)| + |E(P_2)| \leq k + 3 \leq m < j + (m - j + 1)$$

it follows that either P_1 has length $< j$ or P_2 has length $< m - j + 1$, and from the symmetry we may assume the first. By the minimality of k (if u, c_j are nonadjacent, and trivially otherwise) it follows that $d_G(u, c_j) = j - 1$, and therefore P_1 is a shortest path between u, c_j, and it has length $j - 1$ and $u\text{-}P_1\text{-}c_j\text{-}C(c_j, u)\text{-}u$ is a clean shortest odd hole, C' say. Orient C' to agree with the orientation of C on $C(v, u)$. Let u' be the neighbour of c_j in P_1, and let P' be the subpath of P between u', v. Thus P' has length

$$|E(P)| - (|E(P_1)| - 1) = (k + 1) - (j - 2) = k - j + 3.$$

From the minimality of k (applied to C', P') it follows that $d_{C'}(u', v) = k - j + 3$. But $C'(u', v)$ has length $m - j + 1 > k - j + 3$, and $C'(v, u')$ has length strictly longer than P, a contradiction. This proves (6).

Note that c_{2n+1}, c_1, c_2 are all different from c_{m-1}, c_m, c_{m+1}, since $k \geq 2$. From (2), (3) and (6) it follows that the only edges between P^* and $V(C)$ are $p_1 c_1$, $p_k c_m$, possibly one edge from p_1 to one of c_2, c_{2n+1}, and possibly one edge from p_k to one of c_{m-1}, c_{m+1}. If neither or both of the possible extra edges are present, there is an odd hole shorter than C, a contradiction; while if exactly one of the possible extra edges is present, then G contains a pyramid (induced on $V(C \cup P)$), again a contradiction. This proves 4.1. ∎

We use the previous result to prove the following. (This is included just for its simplicity; it is not actually used in the final algorithm.)

4.2 *There is an algorithm with the following specifications:*

Input: *A graph G containing no pyramid or jewel.*

Output: *Determines one of the following:*

    1. G contains an odd hole

    2. there is no clean shortest odd hole in G.

Running time: $O(|V(G)|^4)$.

Proof. Here is an algorithm. For every pair of vertices u, v, find a shortest path $P(u, v)$ between them, if one exists. For every triple u, v, w, test whether the three paths $P(u, v), P(v, w), P(w, u)$ all exist, and if so whether their union is an odd hole. If we find such a hole, output that fact. If not, when all triples have been examined, output that there is no clean shortest odd hole in G.

To see that this works correctly, certainly if the algorithm outputs statement 1 then that is correct. We must check that if statement 2 is false then the algorithm will output statement 1. So assume that statement 2 is false, that is, there is a shortest odd hole C of G that is clean. Choose vertices $u, v, w \in V(C)$, roughly equally spaced in C; more precisely, such that every component of $C \setminus \{u, v, w\}$ contains at most $n - 1$ vertices, where C has length $2n + 1$. Since there is a path joining u, v, the algorithm will find a shortest such path $P(u, v)$. We claim that C can be chosen containing $P(u, v)$. For let L_1 be the path of C joining u, v, not passing through w. Then L_1 has length $\leq n$, from the choice of u, v, w, and so from 4.1, $L_1, P(u, v)$ have the same length. If u, v are adjacent then $P(u, v) = L_1$ and therefore C already contains $P(u, v)$; and otherwise let L_2 be the second path between u, v in C. The union of $L_2, P(u, v)$ is a clean shortest odd hole, by 4.1, and so again we may choose C containing $P(u, v)$. By repeating this for the other two pairs from u, v, w, we see that C can be chosen to include any of $P(u, v), P(v, w), P(w, u)$, and indeed all of them simultaneously. So the union of the three paths joining u, v, w chosen by the algorithm is an odd hole, and therefore in this case the algorithm correctly outputs an odd hole.

The running time of the algorithm as described is $O(|V(G)|^5)$, because after selecting u, v, w and the three paths, it takes quadratic time to check whether the three paths make a hole. With a little more care we can bring it down to $O(|V(G)|^4)$, using the black/white colouring trick we used in 2.2. But here the running time is not crucial, so we omit the details. This proves 4.2. ∎

5 The use of near-cleaners

Let us show next how to use the results of the previous section to complete Routine 1.

5.1 *There is an algorithm with the following specifications:*

Input: *A graph G, containing no pyramid or jewel, and a subset X of $V(G)$.*

Output: *Determines one of the following:*

- *G has an odd hole.*
- *There is no shortest odd hole in C such that X is a near-cleaner for C.*

Running time: $O(|V(G)|^4)$.

Proof. Here is a wasteful way to do it. Enumerate all $Y \subseteq X$ with $|Y| \leq 3$; and apply 4.2 to $G \setminus (X \setminus Y)$ for each such X and Y. If G has an odd hole, then one of these subgraphs has a clean shortest odd hole, and we will therefore detect an odd hole. If not, then X is not a near-cleaner for any shortest odd hole, and we output that.

This is simple to state, but the running time is $O(|V(G)|^7)$, and with more care we can do it more efficiently. So let us do it again.

For every pair $x, y \in V(G)$ of distinct vertices, find a shortest path $R(x, y)$ between x, y with no internal vertex in X, if there is one, and let $r(x, y)$ be its length. If $R(x, y)$ does not exist, let $r(x, y)$

13

be infinite. For all $y_1 \in V(G) \setminus X$ and all 3-vertex paths x_1-x_3-x_2 of $G \setminus y_1$, we check whether all the following are true, where y_2 is the neighbour of y_1 in $R(x_2, y_1)$:

- $r(x_1, y_1)$, $r(x_2, y_1)$ are both finite (and therefore y_2 is defined)

- $r(x_2, y_1) = r(x_1, y_1) + 1 = r(x_1, y_2)$ (= n say)

- $r(x_3, y_1), r(x_3, y_2) \geq n$.

If we find such a choice of x_1, x_2, x_3, y_1, then we output that there is an odd hole. If not, we report that there is no shortest odd hole in C such that X is a near-cleaner for C.

Let us see that the output of this algorithm is correct. First, suppose that there is a choice of x_1, x_2, x_3, y_1 satisfying the three conditions, and let y_2, n be as above. We claim that G contains an odd hole. Let $R(x_1, y_1)$ have vertices p_1-$\cdots$-p_n, and let $R(x_2, y_1)$ have vertices q_1-$\cdots$-q_{n+1}, where $p_1 = x_1, p_n = q_{n+1} = y_1$, $q_1 = x_2$, and $q_n = y_2$. From the definition of $R(x_1, y_1)$ and $R(x_2, y_1)$, none of $p_2, \ldots, p_{n-1}, q_2, \ldots, q_n$ belong to X, and from the choice of y_1, $y_1 \notin X$ (possibly x_1, x_2, x_3 belong to X).

Since $r(x_1, y_1) = r(x_2, y_1) - 1$, it follows that x_2 does not belong to $R(x_1, y_1)$; and for the same reason and since x_1, x_2 are nonadjacent, x_1 does not belong to $R(x_2, y_1)$. Since $r(x_3, y_1), r(x_3, y_2) \geq n$, it follows that x_3 does not belong to $R(x_1, y_1)$ or to $R(x_2, y_1)$, and has no neighbours in $R(x_1, y_1) \setminus x_1$, and none in $R(x_2, y_1) \setminus x_2$. Since $r(x_1, y_2) = n$, y_2 does not belong to $R(x_1, y_1)$. We claim first that $p_2, \ldots, p_{n-1}$ are all different from $q_2, \ldots, q_n$. For suppose that $p_i = q_j$ say, where $2 \leq i \leq n - 1$ and $2 \leq j \leq n$. Then the subpaths of these two paths between p_i, y_1 are both subpaths of shortest paths, and therefore have the same length, that is, $j = i + 1$. So p_1-$\cdots$-p_i-q_{j+1}-$\cdots$-q_n contains a path between x_1, y_2 of length $\leq n - 2$, contradicting that $r(x_1, y_2) = n$. So $R(x_1, y_1)$ and $R(x_2, y_1)$ have no common vertex except y_1. If there are no edges between $R(x_1, y_1) \setminus y_1$ and $R(x_2, y_1) \setminus y_1$, then the union of these two paths and the path x_1-x_3-x_2 is an odd hole, and so the algorithm has performed correctly. So assume that $p_i q_j$ is an edge where $1 \leq i \leq n - 1$ and $1 \leq j \leq n$. We claim $i \geq j$. For if $j = 1$ this is clear, so we assume $j > 1$. There is a path between x_1, y_2 within $\{p_1, \ldots, p_i, q_j, \ldots, q_n\}$, which therefore has length $\leq n - j + i$ and has no internal vertex in X (since $j > 1$); and since $r(x_1, y_2) = n$, it follows that $n - j + i \geq n$, that is, $i \geq j$ as claimed. Consequently $i \geq 2$, since x_1, x_2 are nonadjacent. But also $r(x_2, y_1) \geq n$, and so $j + n - i \geq n$, that is, $j \geq i$. Consequently $i = j$; let us choose i minimum. Then x_3-p_1-$\cdots$-p_i-q_i-$\cdots$-q_1-x_3 is an odd hole, and again the algorithm has answered correctly. So when the algorithm outputs that G has an odd hole, it is true that G has an odd hole.

Now we must prove that when the other output is produced, that also is true. Suppose then that X is a near-cleaner for some shortest odd hole C. Thus for some choice of x_1, x_2, x_3, y_1 above, all these four vertices lie in C and $d_C(x_2, y_1) = d_C(x_1, y_1) + 1$ (= n say), and C is a clean shortest odd hole of length $2n + 1$ in H, where $H = G \setminus (X \setminus \{x_1, x_2, x_3\})$. Note in particular that $V(C) \cap X \subseteq \{x_1, x_2, x_3\}$. We observe:

(1) *If $u, v \in V(C)$, then $r(u, v) \geq d_H(u, v) = d_C(u, v)$.*

For if $R(u, v)$ exists then none of its internal vertices are in X, and so it is a path of H; and consequently $r(u, v) \geq d_H(u, v)$. By 4.1, $d_H(u, v) = d_C(u, v)$. This proves (1).

14

Since $r(x_1, y_1) \leq d_C(x_1, y_1)$ (because $V(C) \cap X \subseteq \{x_1, x_2, x_3\}$), it follows from (1) that $r(x_1, y_1) = n-1$ and $R(x_1, y_1)$ is a shortest path of H between x_1, y_1. By 4.1 we can choose C such that $R(x_1, y_1)$ is a path of C. Similarly $r(x_2, y_1) = n$, and we may assume that $R(x_2, y_1)$ is a path of C. In particular, $y_2 \in V(C)$. By the same argument, $r(x_1, y_2) = d_C(x_1, y_2) = n$. Now $d_C(x_3, y_1) = n$, and so by (1), $r(x_3, y_1) \geq n$, and similarly $r(x_3, y_2) \geq n$. Thus in this case the algorithm would output correctly that G is has an odd hole, and so the output of the algorithm is correct in all cases.

It takes time $O(|V(G)|^3)$ to find all the shortest paths; then we check all quadruples x_1, x_2, x_3, y_1, and each takes constant time. So the time taken here is $O(|V(G)|^4)$, as claimed. This proves 5.1. ∎

6 Some more easily-detectable subgraphs

We turn to the algorithm for Routine 2. We define configurations of type $\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3$ as follows:

1. A *configuration of type* $\mathcal{T}_1$ in G is a hole of length 5.

2. A *configuration of type* $\mathcal{T}_2$ in G is a sequence v_1, v_2, v_3, v_4, P, X such that:

   - v_1-v_2-v_3-v_4 is a path of G
   - X is an anticomponent of the set of all $\{v_1, v_2, v_4\}$-complete vertices
   - P is a path in $G \setminus (X \cup \{v_2, v_3\})$ between v_1, v_4, and no vertex in P^* is X-complete or adjacent to v_2 or adjacent to v_3.

3. A *configuration of type* $\mathcal{T}_3$ in G is a sequence $v_1, \ldots, v_6, P, X$ such that

   - $v_1, \ldots, v_6$ are distinct vertices of G
   - $v_1 v_2, v_3 v_4, v_1 v_4, v_2 v_3, v_3 v_5, v_4 v_6$ are edges, and $v_1 v_3, v_2 v_4, v_1 v_5, v_2 v_5, v_1 v_6, v_2 v_6, v_4 v_5$ are non-edges
   - X is an anticomponent of the set of all $\{v_1, v_2, v_5\}$-complete vertices, and v_3, v_4 are not X-complete
   - P is a path of $G \setminus (X \cup \{v_1, v_2, v_3, v_4\})$ between v_5, v_6, and no vertex in P^* is X-complete or adjacent to v_1 or adjacent to v_2
   - if $v_5 v_6$ is an edge then v_6 is not X-complete.

Clearly we can test whether G contains a configuration of type $\mathcal{T}_1$ in time $O(|V(G)|^5)$, and if so then G is not Berge. We need analogous results for the other two types of configurations.

6.1 *There is an algorithm with the following specifications:*

Input: *A graph G.*

Output: *Reports whether G contains a configuration of type $\mathcal{T}_2$.*

Running time: $O(|V(G)|^6)$.

15

Proof. Here is an algorithm. Enumerate all paths v_1-v_2-v_3-v_4 of G. For each one, find the set Y of all $\{v_1, v_2, v_4\}$-complete vertices. Find the anticomponents of Y. For each anticomponent X, test if there is a path P between v_1, v_4 in $G \setminus \{v_2, v_3\}$, such that no internal vertex of P is adjacent to v_2 or v_3, and no internal vertex of P is X-complete. This evidently performs as claimed. This proves 6.1. ∎

The following is proved in [1]. (In fact it is an easy consequence of 6.5.)

6.2 *Let G be Berge, let X be an anticonnected subset of $V(G)$, and P be an odd path in $G \setminus X$, such that both ends of P are X-complete, and no edge of P is X-complete. Then every X-complete vertex has a neighbour in P^*.*

We deduce:

6.3 *If G contains a configuration of type $\mathcal{T}_2$ then G is not Berge.*

Proof. Let v_1, v_2, v_3, v_4, P, X be a configuration of type $\mathcal{T}_2$ in G. If P is even then v_1-v_2-v_3-v_4-P-v_1 is an odd hole, while if P is odd then since the X-complete vertex v_2 has no neighbour in the interior of P, and the ends of P are X-complete and the internal vertices are not, it follows from 6.2 that G is not Berge. This proves 6.3. ∎

Now let us do the same for configurations of type $\mathcal{T}_3$.

6.4 *There is an algorithm with the following specifications:*

Input: *A graph G.*

Output: *Reports whether G contains a configuration of type $\mathcal{T}_3$.*

Running time: $O(|V(G)|^6)$.

Proof. Here is an algorithm. Enumerate all triples v_1, v_2, v_5 of distinct vertices such that $v_1 v_2$ is an edge and v_5 is nonadjacent to both v_1, v_2. For each choice of v_1, v_2, v_5, find the set Y of all $\{v_1, v_2, v_5\}$-complete vertices, and find its anticomponents. For each anticomponent X, find the maximal connected subset F' containing v_5 with the properties that v_1, v_2 have no neighbours in F' and no vertex of $F' \setminus \{v_5\}$ is X-complete. Let F be the union of F' and the set of all X-complete vertices that are nonadjacent to all of v_1, v_2, v_5 and have a neighbour in F'. Still with the same choice of v_1, v_2, v_5 and X, we enumerate all vertices v_4 that are adjacent to v_1 and not to v_2, v_5, and have a neighbour in F and a nonneighbour in X. For each choice of v_4, we test whether there is a vertex v_3, adjacent to v_2, v_4, v_5 and not to v_1, with a nonneighbour in X. If we find such a vertex v_3, let v_6 be a neighbour of v_4 in F, and P a path from v_6 to v_5 with interior in F'; then $v_1, \ldots, v_6, P, X$ is a configuration of type $\mathcal{T}_3$; output that fact. If after checking all choices of v_1, v_2, v_5, X, v_4 we find no such v_3, then there is no such configuration; output that.

This algorithm evidently tests correctly for configurations of type $\mathcal{T}_3$. To see its running time, there are $O(|V(G)|^3)$ triples v_1, v_2, v_5 to examine. For each, there are linearly many choices of X, and each one takes time $O(|V(G)|^2)$ to process. Then there are linearly many choices of v_4, and each takes linear time to process. So the total running time is $O(|V(G)|^6)$. This proves 6.4. ∎

Next we need to show that no Berge graph contains a configuration of type $\mathcal{T}_3$. We need the following, the Roussel-Rubio lemma [7].

6.5 *Let G be Berge, let X be an anticonnected subset of $V(G)$, and P be an odd path $p_1\text{-}\cdots\text{-}p_n$ in $G \setminus X$ with length ≥ 5, such that p_1, p_n are X-complete and $p_2, \ldots, p_{n-1}$ are not. Then there exist nonadjacent $x, y \in X$ such that there are exactly two edges between x, y and P^*, namely xp_2 and yp_{n-1}.*

We also need:

6.6 *Let G be Berge, and let $X \subseteq V(G)$ be connected. Let $v_1, v_2, v_3, v_4, p_1, p_2$ be distinct vertices of $G \setminus X$, such that*

- v_1v_3 *and* v_2v_4 *are edges, and there are no edges between* $\{v_1, v_3\}$ *and* $\{v_2, v_4\}$

- v_3, v_4 *both have a neighbour in X; no vertex of X is adjacent to both v_3, v_4; and v_1, v_2 have no neighbours in X*

- p_1, p_2 *are nonadjacent; p_1 is adjacent to v_1, v_2, v_3 and not to v_4, and p_2 is adjacent to v_1, v_2, v_4 and not to v_3, and*

- *there is a path between v_3, v_4 with interior in X such that p_1, p_2 have no neighbours in its interior.*

Then p_1, p_2 have no neighbours in X.

Proof. Suppose that one of p_1, p_2 has a neighbour in X. Let Q be a path between v_3, v_4 with interior in X such that p_1, p_2 have no neighbours in its interior. Let $x_1\text{-}\cdots\text{-}x_k$ be a path of vertices of X such that one of p_1, p_2 is adjacent to x_1, and x_k has a neighbour in Q^*, with k minimum. (This exists because Q^* is nonempty and X is connected.) It follows that none of $x_1, \ldots, x_{k-1}$ have neighbours in Q^*, and none of $x_2, \ldots, x_k$ are adjacent to p_1 or p_2. Let A, B be paths from x_1 to v_3, v_4 respectively with interior in $\{x_2, \ldots, x_k\} \cup Q^*$. From the symmetry we may assume that x_1 is adjacent to p_1. Then B can be completed to a hole via $v_4\text{-}v_2\text{-}p_1\text{-}x_1$, and therefore B is odd. Consequently it cannot be completed to a hole via $v_4\text{-}p_2\text{-}v_1\text{-}p_1\text{-}x_1$, and so p_2 has neighbours in $B \setminus \{v_4\}$. Since p_2 has no neighbours in $Q^* \cup \{x_2, \ldots, x_k\}$ it follows that p_2 is adjacent to x_1; and this restores the symmetry between p_1 and p_2. Since B is odd, it cannot be completed to a hole via $v_4\text{-}p_2\text{-}x_1$, and so x_1 is adjacent to v_4. Similarly x_1 is adjacent to v_3, contrary to the last hypothesis of the theorem. This proves 6.6. ∎

These two lemmas are applied to prove the following.

6.7 *If G contains a configuration of type $\mathcal{T}_3$ then G is not Berge.*

Proof. Let G be Berge, and suppose that $v_1, \ldots, v_6, P, X$ is a configuration of type $\mathcal{T}_3$ in G. Let Q be an antipath between v_3, v_4 with interior in X. It can be completed to an antihole via $v_4\text{-}v_5\text{-}v_1\text{-}v_3$, and therefore Q is odd. Hence $v_1\text{-}v_3\text{-}Q\text{-}v_4\text{-}v_2$ is an odd antipath. By 6.5, applied in $\overline{G}$ to the path $v_1\text{-}v_3\text{-}Q\text{-}v_4\text{-}v_2$ and anticonnected set $V(P)$, we deduce that there exist $p_1, p_2 \in V(P)$, adjacent in G, such that the only nonedges between p_1, p_2 and $V(Q)$ are p_1v_4, p_2v_3. By 6.6 applied in $\overline{G}$, we deduce that either some $z \in X$ is nonadjacent to both v_3, v_4, or p_1, p_2 are both X-complete. The first is impossible since $z\text{-}v_1\text{-}v_4\text{-}v_3\text{-}v_5\text{-}z$ is not an odd hole. The second is impossible since no internal vertex of P is X-complete, and if v_5, v_6 are adjacent then by hypothesis v_6 is not X-complete. So G contains no such configuration. This proves 6.7. ∎

In summary, then, we have

6.8 *If G or $\overline{G}$ contains a jewel, pyramid, or configuration of type $\mathcal{T}_1$, $\mathcal{T}_2$ or $\mathcal{T}_3$, then G is not Berge. Moreover, there is an algorithm to test whether G contains such a configuration, with running time $O(|V(G)|^9)$.*

As we shall see in the next two sections, the algorithm of 6.8 is all that we need for Routine 2.

7 Normal subsets

Let C be a graph that is a cycle, and let $A \subseteq V(C)$. An edge of C with both ends in A is called an *A-edge*. An *A-gap* is a subgraph of C composed of a component X of $C \setminus A$, the vertices of A with neighbours in X, and the edges between A and X. (So if some two vertices in A are nonadjacent, the A-gaps are the paths of C of length ≥ 2, with both ends in A and no internal vertex in A.) The *length* of an A-gap is the number of edges in it (so if A consists just of two adjacent vertices, the A-gap has length $|E(C)| - 1$). We speak of cycles, A-gaps, and (A, B)-gaps (defined below) being *odd* or *even* meaning that they have an odd (or even, respectively) number of edges. We say that A is *normal* in C if every A-gap is even.

7.1 *Let C be an odd cycle, and let $A \subseteq V(C)$ be normal. Then there are an odd number of A-edges in C, and consequently $|A| \geq 2$.*

Proof. The edges of C that belong to A-gaps are precisely the edges that are not A-edges. But all A-gaps are even, and C is odd, and so there are an odd number of A-edges. This proves 7.1. ∎

If $A, B \subseteq V(C)$, an (A, B)-*gap* is a path P of C between a, b say, such that a is the unique vertex of P in A and b is the unique vertex of P in B. (Possibly $a = b$ and P has length 0.)

7.2 *Let C be a cycle, and let $A, B \subseteq V(C)$, such that A, B are both normal. Let P be an odd $A \cap B$-gap. Then P includes an A-gap that contains an odd number of B-edges.*

Proof. Since every B-gap is even, and every edge of P that is not a B-edge belongs to a B-gap, it follows that P contains an odd number of B-edges. But every B-edge of P lies in exactly one A-gap contained in P, so one of the A-gaps in P contains an odd number of B-edges. This proves 7.2. ∎

7.3 *Let C be an odd cycle, and let $A, B \subseteq V(C)$ be normal. Let P be an A-gap; then P contains either zero or two (A, B)-gaps. If P contains an odd number of B-edges, then P contains an odd (A, B)-gap and an even one. If P contains an even number of B-edges, then P contains either two odd (A, B)-gaps, or none.*

Proof. By 7.1, $|A|, |B| \geq 2$. Let P have vertices $p_1, \ldots, p_n$ say, in order. Since A is normal and C is odd, it follows that $p_1 \neq p_n$, $p_1, p_n \in A$, and n is odd. If none of $p_1, \ldots, p_n$ belongs to B the claim is true, so we assume that we may choose i, j with $1 \leq i \leq j \leq n$, minimum and maximum respectively such that $p_i, p_j \in B$; and therefore P contains exactly two (A, B)-gaps, namely $p_1\text{-}\cdots\text{-}p_i$ and $p_j\text{-}\cdots\text{-}p_n$. (These are indeed (A, B)-gaps, since either $n < |V(C)|$, or $i < j$, because $|B| \geq 2$.) The sum of their lengths is $n - j + i - 1$, and since n is odd, it follows that exactly one of the two

18

(A, B)-gaps is odd if and only if $j - i$ is odd. It therefore suffices to show that the number of B-edges in P is odd if and only if $j - i$ is odd. But the B-edges in P are precisely the edges between p_i and p_j that do not lie in B-gaps, and every B-gap is even, so an even number of edges of $p_i\text{-}\cdots\text{-}p_j$ are not B-edges. This proves 7.3. ∎

7.4 *Let C be an odd cycle. Let $A_1, \ldots, A_k \subseteq V(G)$ be normal, such that for $1 \leq i < j \leq k$, every (A_i, A_j)-gap is even. Then $A_1 \cap \cdots \cap A_k$ is normal.*

Proof. We proceed by induction on k. If $k = 1$ the result is trivial, so we may assume that $k \geq 2$. Define $A_0 = A_1 \cap A_2$. Since there is no odd (A_1, A_2)-gap, 7.2 and 7.3 imply that A_0 is normal.

(1) *For $3 \leq j \leq k$, every (A_0, A_j)-gap is even.*

For suppose not, and choose a path $p_1\text{-}\cdots\text{-}p_n$ $(= P$ say) of C with n even such that p_1 is the unique vertex of P in A_0, and p_n is the unique vertex of P in A_j. Choose h with $1 \leq h \leq n$ maximum such that $p_h \in A_1$, and i with $1 \leq i \leq n$ maximum such that $p_i \in A_2$. Since every (A_1, A_j)-gap is even, it follows that $n - h$ is even, and therefore h is even, since n is even. Since every A_1-gap is even and $p_1, p_h \in A_1$, it follows that there are an odd number of A_1-edges in the path $p_1\text{-}\cdots\text{-}p_h$. Similarly i is even, and from the symmetry we may assume that $h \leq i$. Now $p_2, \ldots, p_n \notin A_1 \cap A_2$, and in particular no A_1-edge of P is an A_2-edge. Consequently every A_1-edge of P belongs to a unique A_2-gap included in P, and therefore some A_2-gap included in P contains an odd number of A_1-edges. By 7.3, there is an odd (A_1, A_2)-gap, a contradiction. This proves (1).

From (1) and the inductive hypothesis applied to $A_0, A_3, \ldots, A_k$, it follows that $A_0 \cap A_3 \cap \cdots \cap A_k$ is normal, and hence $A_1 \cap \cdots \cap A_k$ is normal. This proves 7.4. ∎

7.5 *Let G be a graph containing no pyramid, and let C be a shortest odd hole in G. Then every C-major vertex has at least four neighbours in C.*

Proof. Let v be a C-major vertex, and suppose it has at most three neighbours in C. Since its set of neighbours in C is normal, it follows from 7.1 that there are an odd number of edges of C with both ends adjacent to v, and therefore there is exactly one such edge; and since v is C-major, it has exactly three neighbours in C. But then the subgraph induced on $V(C) \cup \{v\}$ is a pyramid, a contradiction. This proves 7.5. ∎

We apply the two previous lemmas to show the following. (7.4 and 7.6 were proved by two of us in joint work with Neil Robertson and Robin Thomas, and 7.6 was proved independently by the other three of us in joint work with Michele Conforti and Giacomo Zambelli. Thanks also to Conforti and Zambelli for pointing out an error in an earlier draft of this theorem.).

7.6 *Let G be a graph containing no jewel or pyramid, and let C be a shortest odd hole in G. Let X be a stable set of C-major vertices. Then the set of X-complete vertices in C is normal.*

Proof. Let $X = \{x_1, \ldots, x_k\}$, and for $1 \leq i \leq k$ let A_i be the set of neighbours of x_i in $V(C)$. For $1 \leq i \leq k$, since x_i is C-major and C is a shortest odd hole, it follows that A_i is normal and $|A_i| \geq 4$

by 7.5. If for $1 \leq i < j \leq k$ every (A_i, A_j)-gap is even, then the result follows from 7.4. So we may assume that there is an odd (A_1, A_2)-gap, say P. Let C have vertices $c_1, \ldots, c_{2n+1}$ in order, where P is c_1-$\cdots$-c_r, and r is even, with $2 \leq r \leq 2n$.

(1) *If Q is an even (A_1, A_2)-gap, then $V(P \cap Q) = \emptyset$ and there is an edge between $V(P), V(Q)$.*

For suppose first that $P \cap Q$ is nonempty, say $c_r \in V(Q)$. Then $P \cup Q$ is an odd A_1-gap or A_2-gap, a contradiction. So $P \cap Q$ is empty. Now assume there are no edges between P and Q. Then their union, together with x_1, x_2, forms an odd hole, of length at most that of C. Since C is a shortest odd hole, this new hole has length equal to that of C, and hence Q is the path c_{r+2}-$\cdots$-c_{2n}. Since $|A_1|, |A_2| \geq 4$ it follows that $c_{r+1}, c_{2n+1} \in A_1 \cap A_2$. From the symmetry, we may assume that $c_1 \in A_1$ and $c_r \in A_2$. If $r + 2 = 2n$ then the sequence $c_1, c_{2n+1}, c_{2n}, c_{2n-1}, x_1, P$ is a jewel, a contradiction. So $r + 2 < 2n$; but then c_{r+1}-$\cdots$-c_{2n} is either an odd A_1-gap (if $c_{r+2} \in A_2$) or an odd A_2-gap (if $c_{r+2} \in A_1$), in either case a contradiction. This proves (1).

Now there is an even (A_1, A_2)-gap; for if $A_1 \cap A_2$ is normal, then it is nonempty and there is an (A_1, A_2)-gap of length 0, and otherwise there is an even (A_1, A_2)-gap by 7.2 and 7.3. So by (1) we may assume that c_{r+1}-$\cdots$-c_s is an even (A_1, A_2)-gap, for some odd s with $r + 1 \leq s \leq 2n + 1$. Since $|A_1| \geq 4$, and only two vertices of $P \cup Q$ belong to A_1, it follows that $s \leq 2n - 1$. If both A_1, A_2 meet the path c_{s+2}-$\cdots$-c_{2n}, then there is an (A_1, A_2)-gap contained in this path; it is not even, by (1), since there are no edges between it and P, and it is not odd, by (1), since there are no edges between it and Q, a contradiction. So we may assume that none of $c_{s+2}, \ldots, c_{2n}$ belong to A_1, from the symmetry between A_1, A_2. Since $|A_1| \geq 4$, it follows that $c_{s+1}, c_{2n+1} \in A_1$. Since c_{s+1}-c_{s+2}-$\cdots$-c_{2n+1} is not an odd A_1-gap, it follows that $s = 2n - 1$. Since $|A_2| \geq 4$, it follows that $c_{2n}, c_{2n+1} \in A_2$. But then c_{2n} forms an even (A_1, A_2)-gap, and there are no edges between it and P, contrary to (1). This proves 7.6. ∎

8 Anticonnected sets of C-major vertices

We recall that a hole C of G is *amenable* if

- C is a shortest odd hole in G, of length at least 7, and

- for every anticonnected set X of C-major vertices, there is an X-complete edge in C.

The next result shows that the algorithm of 6.8 provides Routine 2.

8.1 *Let G be a graph containing no pyramid and no configuration of types $\mathcal{T}_1, \mathcal{T}_2$ or $\mathcal{T}_3$, and such that $G, \overline{G}$ both contain no jewel. Then every shortest odd hole in G is amenable.*

Proof. Let C be a shortest odd hole in C. Let us say a subset X of $V(G)$ is *well-behaved* if the set of all X-complete vertices in C is normal. Suppose the result is false, and choose an anticonnected set X of C-major vertices that is not well-behaved, with X minimal. By 7.6, some two vertices in X are adjacent, and therefore there is an antipath of length ≥ 2 contained in X. Let Q be a maximal antipath in X, between $a, b \in X$ say. Then $X \setminus \{a\}, X \setminus \{b\}$ are both anticonnected. Let

A be the set of all $X \setminus \{a\}$-complete vertices in C, and define B similarly. Then from the minimality of X, A and B are normal, and $A \cap B$ is not, since $A \cap B$ is the set of all X-complete vertices in C.

(1) *There exist a vertex in $A \setminus B$ and a vertex in $B \setminus A$ that are nonadjacent.*

For assume that every vertex in $A \setminus B$ is adjacent to every vertex in $B \setminus A$. By 7.2 and 7.3, there is an odd $A \cap B$-gap P, containing an odd (A, B)-gap and an even one. Since every vertex in $A \setminus B$ is adjacent to every vertex in $B \setminus A$ it follows that there is an (A, B)-gap in P of length 1, and one of length 0 (and hence $A \cap B \neq \emptyset$). Let C have vertices $c_1, \ldots, c_{2n+1}$ in order. We may assume that $c_1 \in A \setminus B$ and $c_{2n+1} \in B \setminus A$, and c_1, c_{2n+1} both belong to P^*. Let P be $c_j\text{-}\cdots\text{-}c_{2n+1}\text{-}c_1\text{-}\cdots\text{-}c_i$; thus, $i, j \in \{2, 3, \ldots, 2n\}$ are minimum and maximum respectively such that $c_i, c_j \in A \cap B$. Since P is odd, it follows that $j - i$ is even. Suppose that $i > 2$. Since none of $c_2, \ldots, c_{i-1}$ belong to A (since every vertex in $A \setminus B$ is adjacent to c_{2n+1}), it follows that $c_1\text{-}\cdots\text{-}c_i$ is an A-gap, and therefore i is odd. Consequently $c_{2n+1}\text{-}c_1\text{-}\cdots\text{-}c_i$ is not a B-gap; but every vertex in $B \setminus A$ is adjacent to c_1, and so $c_2 \in B \setminus A$, and no other internal vertex of $c_{2n+1}\text{-}c_1\text{-}\cdots\text{-}c_i$ belongs to B. So $c_2\text{-}\cdots\text{-}c_i$ is an odd path, and since it is not a B-gap it follows that $i = 3$. We have shown then that if $i > 2$ then $i = 3$ and $c_2 \in B \setminus A$. By the same argument, if $j < 2n$ then $j = 2n - 1$ and $c_{2n} \in A \setminus B$. Now not both $c_2 \in B \setminus A$ and $c_{2n} \in A \setminus B$, since every vertex in $A \setminus B$ is adjacent to every vertex in $B \setminus A$; and hence we may assume that $j = 2n$. Since $j - i$ is even it follows that $i \neq 3$, and hence $i = 2$.

There are two cases, depending on whether some vertex of $c_3\text{-}\cdots\text{-}c_{2n-1}$ belongs to $A \cup B$. Suppose first that, say, $c_h \in A$ for some h with $3 \leq h \leq 2n - 1$. Since c_h is not adjacent to c_{2n+1}, it follows that $c_h \in A \cap B$. But then $c_1, c_{2n}, c_2, c_{2n+1}, c_h$ and the antipath $c_1\text{-}a\text{-}Q\text{-}b\text{-}c_{2n+1}$, form a jewel in $\overline{G}$, a contradiction. So none of $c_3, \ldots, c_{2n-1}$ belongs to $A \cup B$. Let Z be the set of all $\{c_2, c_{2n}, c_{2n+1}\}$-complete vertices in G; thus, $X \setminus \{b\} \subseteq Z$. Let Y be the anticomponent of Z that includes $X \setminus \{b\}$. But then $c_{2n}, c_{2n+1}, c_1, c_2$, the path $c_2\text{-}c_3\text{-}\cdots\text{-}c_{2n}$, and Y, form a configuration of type $\mathcal{T}_2$ in G, a contradiction. This proves (1).

(2) *Every vertex of $X \setminus \{a, b\}$ is adjacent to at least one of a, b.*

For suppose that some $x \in X \setminus \{a, b\}$ is nonadjacent to both a, b. Choose $c_i \in A \setminus B$ and $c_j \in B \setminus A$, nonadjacent. Then $x\text{-}c_i\text{-}b\text{-}a\text{-}c_j\text{-}x$ is a configuration of type $\mathcal{T}_1$, a contradiction. This proves (2).

It follows from (2) that Q has length ≥ 3.

(3) $X \setminus \{a, b\}$ *is anticonnected.*

For assume it has more than one anticomponent; then there is an anticomponent Y of $X \setminus \{a, b\}$ disjoint from Q^*. Since X is anticonnected, there exists $y \in Y$ nonadjacent to one of a, b. Since Y is disjoint from Q^* it follows that y is Q^*-complete, and from the maximality of Q it follows that y is nonadjacent to both a, b, contrary to (2). This proves (3).

(4) *Every odd (A, B)-gap has length 1.*

For let C have vertices $c_1, \ldots, c_{2n+1}$ in order, where $c_1\text{-}\cdots\text{-}c_r$ is an odd (A, B)-gap, and assume that $r \geq 3$. We may assume that $c_1 \in A$ and $c_r \in B$. Since $X \setminus \{a, b\}$ is well-behaved, and $c_1\text{-}\cdots\text{-}c_r$

is an odd path and its ends are $X \setminus \{a, b\}$-complete, it follows that some internal vertex of this path is $X \setminus \{a, b\}$-complete, say c_i. But then c_i is nonadjacent to both a, b, since c_1-$\cdots$-c_r is an (A, B)-gap; and therefore a, c_1, c_r, b, c_i, Q is a jewel in $\overline{G}$, a contradiction. This proves (4).

By 7.2 there is an A-gap P containing an odd number of B-edges; and by 7.3, there is an odd (A, B)-gap contained in P. By (4) it has length 1. Consequently we may assume that C has vertices $c_1, \ldots, c_{2n+1}$ in order, where $c_1 \in A \setminus B$ and $c_{2n+1} \in B \setminus A$.

(5) *At least one of $c_3, \ldots, c_{2n-1}$ belongs to $(A \setminus B) \cup (B \setminus A)$.*

For suppose not. Suppose first that none of $c_3, \ldots, c_{2n-1} \in A \cup B$. Since A, B are normal it follows that $c_2 \in A$ and $c_{2n} \in B$. Let R be the path c_2-c_3-$\cdots$-c_{2n}, and let Y be the anticomponent including Q^* of the set of all $\{c_{2n}, c_1, c_2\}$-complete vertices in G. Since $c_2, c_1, c_{2n+1}, c_{2n}, R, Y$ is not a configuration of type $\mathcal{T}_2$, there is a vertex c_h of R^* that is Y-complete. Since no vertex of R^* is $X \setminus \{a\}$-complete, it follows that $X \setminus \{a\} \not\subseteq Y$, and therefore $c_{2n} \notin A$. Similarly $c_2 \notin B$. But c_h is $X \setminus \{a, b\}$-complete since it is Y-complete, and therefore c_h is nonadjacent to a, b since it does not belong to $A \cup B$. But then $a, c_2, c_{2n+1}, b, c_h, Q$ is a jewel in $\overline{G}$, a contradiction.

This proves that there exists h with $3 \leq h \leq 2n-1$ such that $c_h \in A \cup B$, and therefore $c_h \in A \cap B$. If $c_2 \in A \setminus B$, then $a, c_1, c_h, c_{2n+1}, c_2$ and the antipath a-Q-b-c_{2n+1} form a jewel in $\overline{G}$, a contradiction. So $c_2 \notin A \setminus B$, and similarly $c_{2n} \notin B \setminus A$. By (1) it follows that $c_2 \in B \setminus A$ and $c_{2n} \in A \setminus B$. From the symmetry, and since $n > 2$ (because G contains no configuration of type $\mathcal{T}_1$), we may assume that $h \geq 4$. But then a, c_1, c_h, c_2, c_{2n} and the antipath c_2-b-Q-a form a jewel in $\overline{G}$, a contradiction. This proves (5).

(6) *None of $c_3, \ldots, c_{2n-1}$ is $V(Q)$-complete. Consequently, every member of $A \cap B$ has a neighbour in every odd (A, B)-gap.*

For suppose that some c_i is $V(Q)$-complete where $3 \leq i \leq 2n - 1$. From (5) there exists h with $3 \leq h \leq 2n - 1$ such that $c_h \in (A \setminus B) \cup (B \setminus A)$, and from the symmetry we may assume that $c_h \in A \setminus B$. But then $a, c_1, c_i, c_{2n+1}, c_h$ and the antipath c_{2n+1}-b-Q-a, form a jewel in $\overline{G}$, a contradiction. This proves the first claim. For the second, note that since every odd (A, B)-gap has length 1 by (4), it suffices from the symmetry to show that every member of $A \cap B$ has a neighbour in $\{c_{2n+1}, c_1\}$; and since every vertex in $A \cap B$ is $V(Q)$-complete, this follows from the first claim. This proves (6).

(7) $X = V(Q)$.

For from (6), no edge of C is $V(Q)$-complete, and therefore $V(Q)$ is not well-behaved. From the minimality of X it follows that $X = V(Q)$. This proves (7).

(8) *Let P be a path of C containing a vertex of $A \cup B$, such that a, b both have neighbours in P. Suppose that there is an odd (A, B)-gap R such that $V(P), V(R)$ are disjoint and there are no edges between them. Then P includes an odd (A, B)-gap.*

For since R has length 1 by (4), we may assume that R is c_{2n+1}-c_1 and P is a subpath of c_3-c_4-$\cdots$-c_{2n-1}.

22

Choose P minimal such that it contains a vertex of $A \cup B$ and a, b both have neighbours in it. Let its ends be p_1, p_2 say; so one of $p_1, p_2 \in A \cup B$, say $p_1 \in B$, and therefore p_2 is the unique neighbour of b in P. By (6), $p_1 \neq p_2$. Let Y be the anticomponent including Q^* of the set of all $\{c_1, c_{2n+1}, p_1\}$-complete vertices. Since $c_1, c_{2n+1}, a, b, p_1, p_2, P, Y$ do not form a configuration of type $\mathcal{T}_3$, it follows that either some internal vertex of P is Y-complete, or P has length 1 and p_2 is Y-complete. Suppose the first; say c_h is an internal vertex of P, and c_h is Y-complete. It follows that c_h is Q^*-complete, and from the minimality of P, c_h is nonadjacent to both a, b. But $p_1 \in B \setminus A$ by (6), and so a, c_1, p_1, b, c_h, Q is a jewel in $\overline{G}$, a contradiction. So there is no such c_h, and hence P has length 1 and p_2 is Y-complete and therefore Q^*-complete. By (6) p_2 is not $V(Q)$-complete, and since p_2 is adjacent to b, it is therefore nonadjacent to a. But then p_2-p_1 is an odd (A, B)-gap, by (7). This proves (8).

(9) *There is an odd (A, B)-gap in c_3-$\cdots$-c_{2n-1}, and $|A \cap B| \leq 1$.*

For by (5), at least one of $c_3, \ldots, c_{2n-1}$ belongs to $A \cup B$. Since a, b both have at least four neighbours in C by 7.5, and both have a nonneighbour in $\{c_{2n+1}, c_1\}$, they both have neighbours in c_3-$\cdots$-c_{2n-1}. Hence by (8) applied to c_3-$\cdots$-c_{2n-1}, there is an odd (A, B)-gap contained in c_3-$\cdots$-c_{2n-1}. Since it has length 1 by (4), and by (6) every vertex in $A \cap B$ has a neighbour in each odd (A, B)-gap, the claim follows. This proves (9).

(10) *The number of odd (A, B)-gaps in C is odd.*

For every odd (A, B)-gap in C is contained in a unique A-gap; and every B-edge is in a unique A-gap since there is no $A \cap B$-edge by (9). But by 7.3, an A-gap contains an odd number of B-edges if and only if it includes an odd number of odd (A, B)-gaps. Since C contains an odd number of B-edges (since B is normal), it therefore contains an odd number of odd (A, B)-gaps. This proves (10).

Assign an orientation ("clockwise") to C, where c_2 is the successor of c_1. For every odd (A, B)-gap, let its ends be c, d where d immediately follows c in the clockwise order. Then either $c \in A \setminus B$ and $d \in B \setminus A$, or vice versa. If $c \in A \setminus B$, we say c-d is *white* and otherwise it is *black*. From (9), c_{2n+1}-c_1 is not the only odd (A, B)-gap; so by (10) there are two (distinct) successive odd (A, B)-gaps that have the same colour. Hence we may assume that for some i with $1 \leq i \leq 2n$, $c_i \in B \setminus A$ and $c_{i+1} \in A \setminus B$, and the path c_1-$\cdots$-c_i contains no odd (A, B)-gap. Since $c_1 \in A \setminus B$ and $c_i \in B \setminus A$, it follows that $i \geq 2$, and similarly $i + 1 \leq 2n$. Now the path c_1-$\cdots$-c_i contains a neighbour of a, a neighbour of b, and a member of $A \cup B$, and includes no odd (A, B)-gap; and we deduce from (8) that every odd (A, B)-gap contains one of c_{i+1}, c_{2n+1}. By (10) we may assume that $c_{i+2} \in B \setminus A$, and there are exactly three odd (A, B)-gaps in C.

Suppose that $i + 2 \leq 2n$. Since there are no edges between $\{c_1, \ldots, c_{i-1}\}$ and the odd (A, B)-gap $\{c_{i+1}, c_{i+2}\}$, it follows from (8) applied to c_1-c_2-$\cdots$-c_{i-1} that none of $c_1, \ldots, c_{i-1}$ belong to B. So c_{2n+1}-c_1-$\cdots$-c_i is a B-gap, and therefore i is even. Since c_1-$\cdots$-c_i is not an odd (A, B)-gap, it follows that there exists h with $2 \leq h < i$ such that $c_h \in A$. Choose h maximum. Then c_h-$\cdots$-c_{i+1} is an A-gap and c_h-$\cdots$-c_i is an (A, B)-gap, and one of them is odd, a contradiction. This proves that $i = 2n - 1$.

Since A, B are both normal, they both meet c_2-$\cdots$-c_{2n-2}, and yet there are no edges between $\{c_2, \ldots, c_{2n-2}\}$ and the odd (A, B)-gap $\{c_{2n}, c_{2n+1}\}$ and the path includes no odd (A, B)-gap, con-

trary to (8). This proves 8.1. ∎

9 The cleaning algorithm

We turn to Routine 3. For any two distinct vertices a, b, let $N(a, b)$ be the set of all $\{a, b\}$-complete vertices. The definition of an amenable shortest odd hole C implies that cleaning can be done when the set of C-major vertices is anticonnected. How do we proceed when there are at least two anticomponents of C-major vertices? The key idea is as follows. If X is some subset of the C-major vertices, there may or may not be two X-complete vertices in C that have distance at least 3 in C. Smaller sets X tend to have this property, and larger sets tend not to. In both cases a good thing happens. First, if X does have two such common neighbours (a, b say) then all members of $N(a, b)$ are C-major, and this set includes X; so we could easily guess a set of C-major vertices including X, just by trying all pairs a, b and outputting the set $N(a, b)$ for each pair. Thus in this case, X is essentially easily "guessable". On the other hand, if X does not have two such common neighbours a, b, then the set of all X-complete vertices has very limited intersection with C, and yet contains "most" C-major vertices (all of them except those in the anticomponents of C-major vertices that meet X); and therefore if only we could guess X, we could guess most C-major vertices. If we choose X right on the border, maximal such that a, b both exist, and add one more C-major vertex to it, we essentially get both good things at once. This is still not quite enough to do cleaning; the last trick is not just to maximize X, but to lexicographically maximize the sizes of the anticomponents of X. Then that works, as we shall see. Let us explain the details.

A triple (a, b, c) of vertices is *relevant* if a, b are distinct and nonadjacent, and $c \notin N(a, b)$ (possibly $c \in \{a, b\}$). For every relevant triple (a, b, c), we make the following definitions:

- $r(a, b, c)$ is the cardinality of the largest anticomponent of $N(a, b)$ that contains a nonneighbour of c (or 0, if c is $N(a, b)$-complete)

- $Y(a, b, c)$ is the union of all anticomponents of $N(a, b)$ that have cardinality strictly greater than $r(a, b, c)$

- $W(a, b, c)$ is the anticomponent of $N(a, b) \cup \{c\}$ that contains c

- $Z(a, b, c)$ is the set of all $Y(a, b, c) \cup W(a, b, c)$-complete vertices, and

- $X(a, b, c) = Y(a, b, c) \cup Z(a, b, c)$.

The algorithm depends on the following lemma.

9.1 *Let C be a shortest odd hole in G, with length at least 7. Then there is a relevant triple (a, b, c) of vertices such that:*

- *the set of all C-major vertices not in $X(a, b, c)$ is anticonnected, and*

- *$X(a, b, c) \cap V(C)$ is a subset of the vertex set of some 3-vertex path of C.*

Proof. Let M be the set of all C-major vertices. Choose vertices $a, b \in V(C)$, such that both paths of C joining them have length at least 3 (this is possible since C has length at least 7). Let the anticomponents of $N(a, b)$ have cardinalities $n_1, \ldots, n_k$ in non-increasing order. Choose a, b such that n_1 is as large as possible, and subject to that n_2 is as large as possible, and so on. We observe first that $N(a, b) \subseteq M$, and that for any vertex $c \notin N(a, b)$, (a, b, c) is relevant and $Y(a, b, c)$ is disjoint from $V(C)$.

Suppose first that $M \subseteq N(a, b)$. Then equality holds. Moreover, $r(a, b, a) = 0$, and $Y(a, b, a) = N(a, b)$ and so every C-major vertex belongs to $X(a, b, a)$. But $W(a, b, a) = \{a\}$, and therefore $Z(a, b, a) \cap V(C)$ is a subset of the set of neighbours of a in C; and consequently, $X(a, b, c) \cap V(C)$ is a subset of a 3-vertex path of C. In this case the triple (a, b, a) satisfies the theorem.

We may therefore assume that there exists $c \in M \setminus N(a, b)$. For any such vertex c, (a, b, c) is relevant; choose c such that $r(a, b, c)$ is as large as possible. We claim that (a, b, c) satisfies the theorem.

(1) $W(a, b, c) \subseteq M \setminus X(a, b, c)$; and every vertex of $M \setminus X(a, b, c)$ either belongs to $W(a, b, c)$ or has a nonneighbour in $W(a, b, c)$. Consequently $M \setminus X(a, b, c)$ is anticonnected.

For $W(a, b, c)$ is disjoint from $Y(a, b, c), Z(a, b, c)$ from the definition of these sets, and so $W(a, b, c) \subseteq M \setminus X(a, b, c)$. For the second assertion, let $v \in M \setminus X(a, b, c)$. We claim that v is $Y(a, b, c)$-complete. For if $v \in N(a, b)$ then this is true, since $v \notin Y(a, b, c)$ and $Y(a, b, c)$ is a union of anticomponents of $N(a, b)$. If $v \notin N(a, b)$ then since $v \in M$, it follows that v is $Y(a, b, c)$-complete from the choice of c. This proves that v is $Y(a, b, c)$-complete. Since $v \notin Z(a, b, c)$, it follows that either $v \in W(a, b, c)$ or v has a nonneighbour in $W(a, b, c)$. This proves (1).

(2) $X(a, b, c) \cap V(C)$ is a subset of the vertex set of some 3-vertex path of C.

For suppose not. Since C has length at least 7, there exist $a', b' \in Z(a, b, c) \cap V(C)$ with distance at least 3 in C. Then $Y(a, b, c) \cup W(a, b, c) \subseteq N(a', b')$, and so every anticomponent of $N(a, b)$ with cardinality strictly greater than $r(a, b, c)$ is a subset of an anticomponent of $N(a', b')$. But also $W(a, b, c)$ has cardinality strictly greater than $r(a, b, c)$, from the definition of $r(a, b, c)$, and $W(a, b, c)$ is not an anticomponent of $N(a, b)$, and $W(a, b, c)$ is a subset of an anticomponent of $N(a', b')$. It follows that replacing a, b by a', b' increases lexicographically the sequence $n_1, \ldots, n_k$, contrary to the choice of a, b. This proves (2).

From (1) and (2), it follows that (a, b, c) satisfies the theorem. This proves 9.1. ∎

The algorithm for Routine 3 is as follows.

9.2 *There is an algorithm with the following specifications:*

Input: *A graph G.*

Output: *$O(|V(G)|^5)$ subsets of $V(G)$, such that if C is an amenable hole in G, then one of the subsets is a near-cleaner for C.*

Running time: *$O(|V(G)|^5)$.*

Proof. Here is the algorithm. For every two adjacent vertices u, v, compute the set $N(u, v)$, and list all such sets. For each relevant triple (a, b, c), compute the set $X(a, b, c)$, and list all such sets. Output all subsets that are the union of a set from the first list and one from the second. This completes the algorithm.

To see that this output is correct, suppose that C is an amenable hole in G. By 9.1, there is a relevant triple (a, b, c) satisfying that theorem. Since the set (T, say) of all C-major vertices not in $X(a, b, c)$ is anticonnected, and C is amenable, there is an edge uv of C that is T-complete; and therefore $T \subseteq N(u, v)$. But then $N(u, v) \cup X(a, b, c)$ is a near-cleaner for C, and it is one of the sets in the output. The running time is evidently as claimed. ∎

10   The algorithm for Bergeness

Let us put these pieces together. The main result of the paper is the following:

10.1 *There is an algorithm with the following specifications:*

Input: *A graph G.*

Output: *Determines whether G is Berge.*

Running time: $O(|V(G)|^9)$.

Proof. First, use the algorithm of 6.8 to test whether one of $G, \overline{G}$ contains a jewel, a pyramid, or a configuration of type $\mathcal{T}_1, \mathcal{T}_2$ or $\mathcal{T}_3$. If so, we output that G is not Berge and stop. If not, then by 8.1, every shortest odd hole in G is amenable. We run 9.2, and obtain the $O(|V(G)|^5)$ subsets. For each subset X in turn, we run 5.1 on the pair G, X. If we find that G has an odd hole, we output that and stop. If after examining all the sets X, we still have not found an odd hole, we turn to $\overline{G}$, and run the same procedure on that. (There is no need to repeat the algorithm of 6.8, of course.) If again we do not detect an odd hole, we report that G is Berge. That completes the description of the algorithm.

Let us show that the output is correct. We must show that G is not Berge if and only if the algorithm reports that G is not Berge. From the construction of the algorithm, if the algorithm reports that G is not Berge, then this is true. For the converse, suppose that G is not Berge. Since we run the same algorithm on G and on $\overline{G}$, we may assume that there is an odd hole in G, by replacing G by $\overline{G}$ if necessary. Hence there is a shortest odd hole in G, say C. We may assume that in the call of 6.8, we did not detect that G is not Berge, and therefore G and $\overline{G}$ both contain no jewel, pyramid, or configuration of type $\mathcal{T}_1, \mathcal{T}_2$ or $\mathcal{T}_3$. Hence by 8.1, C is amenable. Thus the call of 9.2 functions as it should, and one of the subsets X it outputs is a near-cleaner for C. Therefore when we apply 5.1 to the pair G, X, the algorithm will discover an odd hole and report that. Thus the output is correct in all cases.

Finally, let us add up the total running time. The call to 6.8 takes time $O(|V(G)|^9)$. Running 9.2 takes time $O(|V(G)|^5)$, and then for each of $O(|V(G)|^5)$ subsets we have to call 5.1, each call taking time $O(|V(G)|^4)$. Then we repeat on the complement. The whole running time is $O(|V(G)|^9)$, as claimed. This proves 10.1. ∎

11 Appendix

Here we give another algorithm for Routine 1. First, by making use of 2.2 and 3.1, we may assume that the input graph contains no pyramid or jewel. In the version of Routine 1 given earlier, we then call 4.2 or its more efficient variant 5.1, but there is an alternative method that can be applied at this stage, that can be shown to work using just a special case of 4.1. What is presented in this appendix is the work of CLV, although an almost identical algorithm was developed independently by ChS in joint work with Neil Robertson and Robin Thomas.

Let us say a graph G is *clean* if it is either odd-hole-free or it contains a clean shortest odd hole. We will present an algorithm that takes as input a clean graph G containing no pyramid or jewel, and recognizes whether G is odd-hole-free. The idea is to decompose the input graph G into a polynomial number of simpler graphs $G_1, \ldots, G_m$ so that the following two properties are satisfied:

(1) G is odd-hole-free if and only if G_i is odd-hole-free for every $i = 1, \ldots, m$, and

(2) for every $i = 1, \ldots, m$ it is easy to check directly whether G_i is odd-hole-free.

The basis of this recognition algorithm is the following decomposition theorem for odd-hole-free graphs by Conforti, Cornuéjols and Vušković [3]. For Berge graphs, this result also follows from the decomposition theorem of Chudnovsky, Robertson, Seymour and Thomas [1].

A set S of vertices is a *double star* if S contains two adjacent vertices u and v such that $S \subseteq N(u) \cup N(v)$. Here $N(x)$ denotes the set of vertices adjacent to vertex x. We say that S is *centred at uv*. The vertex set S is a *cutset* of G if $G \setminus S$ contains more connected components than G.

A graph G has a *2-join* $V_1 | V_2$ with *special sets* (A_1, A_2, B_1, B_2) if its vertices can be partitioned into sets V_1 and V_2 so that, for $i = 1, 2$, V_i contains disjoint, nonempty vertex sets A_i and B_i, such that every vertex of A_1 is adjacent to every vertex of A_2, every vertex of B_1 is adjacent to every vertex of B_2, and there are no other adjacencies between V_1 and V_2. Furthermore, for $i = 1, 2$, $|V_i| > 2$ and the graph induced by V_i is not a path. 2-joins were introduced by Cornuéjols and Cunningham [5] in a special case and by Conforti, Cornuéjols, Kapoor and Vušković [2] in the above form.

A *basic* graph is a bipartite graph or the line graph of a bipartite graph or the complement of a line graph of a bipartite graph.

11.1 *[3] If G is an odd-hole-free graph, then either G is basic, or G has a double star cutset or a 2-join.*

Checking whether a graph is basic can easily be done in polynomial time [6, 8]. This is an answer to (2) above. A polynomial algorithm for finding a 2-join is given in [2]. It is not difficult to find a double star cutset in polynomial time: For any two adjacent vertices u, v and any two nonadjacent vertices x, y, test whether there is a double star cutset centred at u and v that disconnects x and y by removing all the neighbours of u and all the neighbours of v except x and y, and checking whether x and y belong to distinct connected components of the resulting graph. A crude implementation runs in time $O(|V(G)|^6)$. Therefore the main difficulty in applying Theorem 11.1 is to decompose a graph G that has a double star cutset or a 2-join into "blocks of decomposition" G_i that satisfy (1) above.

Blocks of decomposition

We now define the blocks of decomposition for double star cutsets and 2-joins. Remember that our goal is to satisfy (1) above. If $X \subseteq V(G)$ then we denote the subgraph of G induced on X by $G[X]$.

2-Join Decomposition: Let $V_1|V_2$ be a 2-join of G with special sets (A_1, A_2, B_1, B_2). If there does not exist a path from a vertex of A_2 to a vertex of B_2 in $G[V_2]$ then we define *block* G_1 to be the subgraph of G induced by $V_1 \cup \{a_2, b_2\}$, where $a_2 \in A_2$ and $b_2 \in B_2$. Otherwise, let Q_2 be a shortest path from A_2 to B_2 in $G[V_2]$. We define *block* G_1 to be the graph obtained from $G[V_1 \cup V(Q_2)]$ by replacing Q_2 by a path P_2 of length 4 if Q_2 is of even length, and of length 5 otherwise. Path P_2 is called the *marker path.* Block G_2 is defined similarly.

11.2 *Let G_1 and G_2 be the blocks of a 2-join decomposition of G. Then G is odd-hole-free if and only if G_1 and G_2 are odd-hole-free. Furthermore, if G contains a clean odd hole of length strictly greater than 5, then G_1 or G_2 contains a clean odd hole of length strictly greater than 5.*

Proof. Assume first that G is odd-hole-free and that G_1 contains an odd hole C. Suppose that C does not contain the marker path P_2. Since C is not a hole of G, it contains both endvertices of Q_2 and Q_2 is an edge. But then G contains a shorter odd hole, a contradiction. Therefore C contains the marker path P_2. Let $C_1 = C \cap V_1$. Then $C_1 \cup Q_2$ induces an odd hole of G, a contradiction.

Next we prove that if G contains an odd hole C, then G_1 or G_2 contains an odd hole. If C does not contain a vertex in each of A_1, A_2, B_1, B_2, then C is contained in G_1 or G_2. Thus we can assume that C contains a vertex in each of A_1, A_2, B_1, B_2. First assume that C contains exactly four vertices of $A_1 \cup A_2 \cup B_1 \cup B_2$. Let $C_1 = C \cap V_1$ and $C_2 = C \cap V_2$. Since C is odd, we may assume without loss of generality that C_1 is even and C_2 is odd. If P_2 is odd then $C_1 \cup P_2$ is an odd hole of G_1. Hence we can assume that P_2 is even. Then one of $P_1 \cup Q_2$ or $P_1 \cup C_2$ induces an odd hole in G_2. Now assume that C contains more than four vertices of $A_1 \cup A_2 \cup B_1 \cup B_2$. W.l.o.g. C contains two vertices $a_1, a'_1 \in A_1$. Then C contains exactly one vertex in A_2, which is adjacent to a_1 and a'_1. Since C is a hole it cannot contain more than one vertex in B_2. Hence C is entirely contained in G_1.

The second statement of the theorem follows by observing that if the odd hole C defined in the previous paragraph is a clean odd hole of length strictly greater than 5, then the odd hole found in G_1 or G_2 is also a clean odd hole of length strictly greater than 5 (for example the fact that $P_1 \cup Q_2$ is clean follows from the fact that Q_2 is a shortest path from A_2 to B_2 in $G[V_2]$). This proves 11.2. ∎

Double Star Decomposition: Let S be a double star cutset of G and $H_1, H_2, \ldots, H_n$ the connected components of $G \backslash S$. We define the *blocks* of the decomposition to be the graphs $G_1, \ldots, G_n$ where $G_i = G[V(H_i) \cup S]$.

This definition of blocks for the double star cutset does not preserve the odd-hole-free property. Consider a graph G that consists of a 5-hole $C = x_1, x_2, x_3, x_4, x_5, x_1$ and a vertex x adjacent to x_1, x_2 and x_4. If we decompose G with a double star cutset $N(x) \cup \{x\}$ then neither of the blocks contains an odd hole. In the subsection below entitled "Double star decomposition" we show how to preserve the odd-hole-free property if the input graph G is clean.

Clean holes

In this subsection we show that if a shortest odd hole C^* in a graph G is clean, then the entire family of shortest odd holes obtained from C^* through certain vertex and edge substitutions is also clean

in G.

For an odd hole C, a vertex $u \in V(G) \setminus V(C)$ is called C-*minor* if it is not C-major. Consider a C-minor vertex u and let P be the subpath of C of length at most two (P possibly empty) such that $N(u) \cap V(C) \subseteq V(P)$ and u is adjacent to both endvertices of P. We say that u is a C-minor vertex of Type i if $i = |V(P)|$.

Let C be an odd hole and u a C-minor vertex of Type 3, with neighbours in C contained in a subpath u_1, u_2, u_3 of C. Let C' be the hole induced by $(V(C) \setminus \{u_2\}) \cup \{u\}$. We say that C' is obtained from C through a *minor vertex substitution*. Note that C and C' have the same length.

A C-*minor edge* is an edge uv such that both u and v are C-minor vertices, and for some $u'v'$-subpath P of C of length three, $(N(u) \cup N(v)) \cap V(C) \subseteq V(P)$, and u is adjacent to u', and v is adjacent to v'. Note that u is not adjacent to v' and v is not adjacent to u'. Let C' be the hole induced by $(V(C) \setminus V(P)) \cup \{u, v, u', v'\}$. We say that C' is obtained from C through a *minor edge substitution*. Note that C and C' have the same length.

Let C be an odd hole in a graph G. We define $\mathcal{S}_G(C)$ to be the family of all holes of G obtained from C through a sequence of minor vertex substitutions or minor edge substitutions.

The next result is a special case of 4.1. However, we only need the special case of 4.1 when $d_G(u, v) \leq 3$, and for that case the proof of 4.1 can be considerably shortened (to about a page). There is no other application of 4.1 in the algorithm described in this appendix.

11.3 *Let G be a graph containing a shortest odd hole C^* but no jewel nor pyramid. If C^* is clean then all holes in $\mathcal{S}_G(C^*)$ are clean.*

Double star decomposition

In this subsection we decompose clean graphs with double star cutsets.

11.4 *Let G be a graph that contains a shortest odd hole C^*, but no jewel nor pyramid. If $u, v \in V(G) \setminus V(C^*)$ are two adjacent C^*-minor vertices then one of the following is true.*

(i) *uv is a C^*-minor edge.*

(ii) *The vertices of $(N(u) \cup N(v)) \cap V(C^*)$ are contained in a subpath P of C^* of length at most two, and if P is of length 2 then u or v is a C^*-minor vertex of Type 3.*

Proof. Let P be a shortest subpath of C^* such that the vertices of $(N(u) \cup N(v)) \cap V(C^*)$ are contained in P. Suppose that P is of length 2. If neither u nor v is a C^*-minor vertex of Type 3, $V(C^*) \cup \{u, v\}$ induces a jewel. Therefore, (ii) holds. If P is of length 3 then uv is a C^*-minor edge and therefore (i) holds. Now we assume that P is of length strictly greater than 3. If exactly one of u and v is a C^*-minor vertex of Type 2, then there is a pyramid. If u and v are both of Type 3 and they have a common neighbour in C^*, then there is a jewel. In all other cases, $G[V(C^*) \cup \{u, v\}]$ contains an odd hole shorter than C^*. This proves 11.4. ∎

11.5 *Let G be a graph that contains a clean shortest odd hole C^*, but does not contain a jewel or a pyramid. If S is a double star cutset of G, then some hole of $\mathcal{S}_G(C^*)$ is entirely contained in one of the blocks of the decomposition by S.*

Proof. Let S be centred at uv, and suppose that C^* is not entirely contained in one block of the decomposition. Then C^* does not contain both u and v. Suppose that C^* contains u, but not v. Since C^* is clean and is contained in no block of the decomposition, v is a C^*-minor vertex of Type 3. Hence the hole obtained by substituting v into C^* is in $\mathcal{S}_G(C^*)$ and entirely contained in one block of the decomposition. So we may assume that C^* contains neither u nor v. Then by 11.4, one of the holes in $\mathcal{S}_G(C^*)$ is entirely contained in one block of the decomposition. This proves 11.5. ∎

11.6 *There is an algorithm with the following specifications:*

Input: *A connected clean graph G that does not contain a jewel, a pyramid, a 5-hole or a 7-hole.*

Output: *A family $\mathcal{L}$ of induced subgraphs of G that satisfies the following properties:*

   (1) G is odd-hole-free if and only if all the graphs in $\mathcal{L}$ are odd-hole-free.

   (2) The graphs in $\mathcal{L}$ do not have a double star cutset.

   (3) The number of graphs in $\mathcal{L}$ is $O(|V(G)|^2)$.

Running Time: $O(|V(G)|^8)$.

Proof. The algorithm is as follows. Initialize $\mathcal{L} = \emptyset$ and $\mathcal{L}' = \{G\}$, and perform the following iterative step: If $\mathcal{L}' = \emptyset$ then stop. Otherwise, remove a graph F from $\mathcal{L}'$. If the distance between every pair of vertices of F is strictly less than 4 in G, discard F and iterate. Otherwise, if F has no double star cutset, then add F to $\mathcal{L}$ and iterate. Otherwise, let S be a double star cutset in F, construct the blocks of the decomposition by S, add them to $\mathcal{L}'$ and iterate.

(2) holds by the construction of the algorithm. We now show that (1) holds. Since the graphs in $\mathcal{L}$ are induced subgraphs of G, if G is odd-hole-free then all graphs in $\mathcal{L}$ are odd-hole-free. Suppose G contains a clean shortest odd hole C^*. Note that by 11.3 all holes in $\mathcal{S}_G(C^*)$ are clean. Since $C \in \mathcal{S}_G(C^*)$ is of length greater than 7, it contains two vertices u and v that are at distance at least 4 in C. By 11.4 u and v are at distance at least 4 in G as well. Hence by 11.5, some graph in $\mathcal{L}$ contains an odd hole of $\mathcal{S}_G(C^*)$.

We prove (3) by showing that the number of graphs in $\mathcal{L}$ is bounded by the number of pairs of vertices at distance at least 4 in G. Let S be a double star cutset of a graph F, and let $F_1, \ldots, F_m$ be the blocks of the decomposition. Let u and v be two vertices of F that are at distance at least 4 in G (and hence in F). The pair of vertices $\{u, v\}$ cannot be contained in two different blocks of the decomposition since otherwise they would both have to be in S, but since S is a double star, all vertices of S are at distance at most 3. Therefore no pair of vertices that are at distance at least 4 in G can be contained in different graphs in $\mathcal{L}$.

Finding a double star cutset and constructing blocks of decomposition can be done in time $O(|V(G)|^6)$. This is performed at most $O(|V(G)|^2)$ times, giving $O(|V(G)|^8)$ time complexity. This proves 11.6. ∎

2-join decomposition

In this subsection we decompose a clean graph that has no double star cutset using 2-join decompositions, without creating any new double star cutset.

11.7 *If a graph G has a 2-join $V_1|V_2$ with special sets (A_1, A_2, B_1, B_2) such that $V_1 \setminus (A_1 \cup B_1) = \emptyset$ and $V_2 \setminus (A_2 \cup B_2) = \emptyset$ then G contains no clean odd hole of length strictly greater than five.*

Proof. Suppose that C is a clean odd hole of length at least seven in G. Since there is no C-major vertex, C cannot be entirely contained in $A_1 \cup B_1$ or $A_2 \cup B_2$. Now we assume with loss of generality that C contains one vertex of A_2 and two vertices of A_1. Then C contains at least two vertices of B_1 but no vertex of B_2. Now any vertex of B_2 is C-major, a contradiction. This proves 11.7. ∎

11.8 *Suppose that a connected graph G has a 2-join $V_1|V_2$ with special sets (A_1, A_2, B_1, B_2) such that at least one of $V_1 \setminus (A_1 \cup B_1)$ and $V_2 \setminus (A_2 \cup B_2)$ is nonempty. Let G_1 and G_2 be the blocks of a 2-join decomposition of G. If G does not have a double star cutset then the following hold.*

(1) Both $V_1 \setminus (A_1 \cup B_1)$ and $V_2 \setminus (A_2 \cup B_2)$ are nonempty.

(2) For $i = 1, 2$, G_i does not have a double star cutset.

(3) For $i = 1, 2$, $|V_i| \geq 6$.

Proof. For $i = 1, 2$ let $\mathcal{P}_i$ be the set of all paths in $G[V_i]$ with one vertex in A_i, the other in B_i and no intermediate vertex in $A_i \cup B_i$. (1) follows from the following claim.

Claim 1: For $i = 1, 2$, $\mathcal{P}_i \neq \emptyset$ and all paths of $\mathcal{P}_i$ are of length at least 2.

Proof of Claim 1: Let $u \in A_1$ and $v \in B_1$. If $\mathcal{P}_1 = \emptyset$ then, since $|V_1| > 2$, either $\{u\} \cup A_2$ or $\{v\} \cup B_2$ is a double star cutset of G. So $\mathcal{P}_1 \neq \emptyset$ and similarly $\mathcal{P}_2 \neq \emptyset$. Now suppose that uv is an edge. If $V_2 \setminus (A_2 \cup B_2) \neq \emptyset$ then $\{u, v\} \cup A_2 \cup B_2$ is a double star cutset of G. So $V_2 \setminus (A_2 \cup B_2) = \emptyset$. Since $\mathcal{P}_2 \neq \emptyset$ there is an edge from A_2 to B_2, and hence $V_1 \setminus (A_1 \cup B_1) = \emptyset$. But this contradicts the assumption that at least one of $V_1 \setminus (A_1 \cup B_1)$ and $V_2 \setminus (A_2 \cup B_2)$ is nonempty. This completes the proof of Claim 1.

To prove (2) in 11.8, we may assume without loss of generality that G_1 has a double star cutset S centred at xy. By Claim 1, G_1 contains the marker path $P_2 = a_2, \ldots, b_2$. First suppose that $x, y \in V_1$. By Claim 1 no vertex of A_1 is adjacent to B_1, so S cannot contain both a_2 and b_2. W.l.o.g. S does not contain b_2. If S contains a_2 then $S \cup A_2$ is a double star cutset of G. So S does not contain a_2, and hence S is a double star cutset of G. So x or y is in P_2. By Claim 1, $\mathcal{P}_1 \neq \emptyset$ and hence S must contain a_2 or b_2. Since P_2 is of length 4 or 5, S cannot contain both a_2 and b_2. W.l.o.g. assume that S contains a_2 but not b_2. Suppose that neither x nor y coincides with a_2. Since $\mathcal{P}_1 \neq \emptyset$, there is a connected component of $G_1 \setminus S$ that contains $B_1 \cup \{b_2\}$ and some vertex $u \in A_1$. Since G is connected, some vertex of $A_1 \setminus \{u\}$ is contained in another connected component of $G_1 \setminus S$. Then $\{u\} \cup A_2$ is a double star cutset of G. Thereforei we may assume that $x = a_2$. By the above argument we may also assume that S contains a vertex of A_1. In fact, without loss of generality we may assume that $y \in A_1$. But then $(S \setminus V(P_2)) \cup A_2$ is a double star cutset of G.

To prove (3) let Q be a shortest path in $\mathcal{P}_2$. By Claim 1, Q is of length at least 2. If Q is of length at least 4, then by definition of 2-join, $|V_2| \geq 6$. So we may assume that Q is of length 2 or 3. By definition of 2-join, there exists $w \in V_2 \setminus V(Q)$. If $|A_2| = |B_2| = 1$ then $V(Q)$ is a double star cutset of G that separates w from V_1. So we may assume that $w \in A_2$. Let $Q = x_1, \ldots, x_k$, where $x_1 \in A_2$ and $x_k \in B_2$. Let $S = (N(x_1) \cup N(x_2)) \setminus \{w\}$. Since S cannot be a double star cutset of G, there exists a path P from w to B_2 in $G \setminus S$. By Claim 1, P contains at least 3 vertices, and hence $|V_2| \geq 6$. Similarly, $|V_1| \geq 6$. This proves 11.8. ∎

11.9 *There is an algorithm with the following specification:*

Input: *A connected clean graph G that has no double star cutset and no hole of length 5.*

Output: *Either an odd hole of G, or a family $\mathcal{L}$ of graphs that satisfies the following properties:*

      (1) G is odd-hole-free if and only if all graphs in $\mathcal{L}$ are odd-hole-free.

      (2) No graph of $\mathcal{L}$ has a double star cutset or a 2-join.

      (3) The number of graphs in $\mathcal{L}$ is $O(|V(G)|)$.

Running Time: $O(|V(G)|^8)$.

Proof. The algorithm is as follows. Initialize $\mathcal{L} = \emptyset$ and $\mathcal{L}' = \{G\}$, and perform the following iterative step. If $\mathcal{L}' = \emptyset$ then stop. Otherwise, remove a graph F from $\mathcal{L}'$. If F has no 2-join, then add F to $\mathcal{L}$ and iterate. Otherwise, let $V_1|V_2$ be a 2-join of F with special sets (A_1, A_2, B_1, B_2). If $V_1 \setminus (A_1 \cup B_1) = \emptyset$ and $V_2 \setminus (A_2 \cup B_2) = \emptyset$, discard F and iterate. Otherwise, construct the blocks of the 2-join decomposition of F, say F_1 and F_2. For $i = 1$ or 2, if $|V_i| \leq 7$, check directly whether F_i contains an odd hole. If it does, output this result and otherwise discard F_i. If $|V_i| > 7$, add F_i to $\mathcal{L}'$. Iterate.

(1) follows from 11.2 and 11.7, and (2) follows from 11.8.

To prove that $|\mathcal{L}|$ is $O(|V(G)|)$, we construct a decomposition tree T whose root is G and whose leaves are the graphs in $\mathcal{L}$. Let F be a nonleaf vertex of T, let $V_1|V_2$ be a 2-join of F with special sets (A_1, A_2, B_1, B_2), and let F_1, F_2 be the two blocks of the 2-join decomposition of F. Note that $V_i \neq A_i \cup B_i$ for $i = 1, 2$ by 11.8. We define $\phi(F) = |V(F)| - 12$, $\phi(F_i) = |V(F_i)| - 12$ for $i = 1, 2$. Assume first that both F_1 and F_2 appear in T. Since only F_i with $|V_i| > 7$ is added to $\mathcal{L}'$ and the marker path contains at least five vertices, $\phi(F_i) \geq 1$ for $i = 1, 2$. Furthermore, $\phi(F_1) + \phi(F_2) \leq \phi(F)$ by the fact that the marker path contains at most six vertices. Now assume that only one of the blocks F_1, F_2 belongs to T, say F_1. Then $\phi(F_1) \leq \phi(F)$ since $|V_2| \geq 6$ by 11.8 and the marker path of F_1 contains at most six vertices. Let $B_1, \ldots, B_k$ be the leaves of T. Then $k \leq \Sigma_{i=1}^{k} \phi(B_i) \leq \phi(G) = |V(G)| - 12$. This implies that the number of leaves in T is $O(|V(G)|)$.

Finding a 2-join takes time $O(|V(G)|^7)$ using the crude implementation in [2], and this algorithm is applied at most $O(|V(G)|)$ times, which yields an overall complexity of $O(|V(G)|^8)$. This proves 11.9. ∎

Recognition algorithm for odd-hole-free clean graphs

As explained earlier, it suffices to show how to handle graphs with no pyramid or jewel, and we also may assume that they are connected and contain no 5-hole or 7-hole. Then we use the following.

11.10 *There is an algorithm with the following specification:*

Input: *A connected clean graph G, that contains no pyramid, jewel, 5-hole or 7-hole.*

Output: *ODD-HOLE-FREE when G is odd-hole-free, and NOT ODD-HOLE-FREE otherwise.*

Running Time: $O(|V(G)|^{10})$.

Proof. The algorithm has three steps. In Step 1, we apply the Double Star Decomposition Algorithm to G, and let $\mathcal{L}_1$ be the output family of graphs. In Step 2, we set $\mathcal{L}_2 = \emptyset$. For every graph in $\mathcal{L}_1$ apply the 2-Join Decomposition Algorithm. If the output of this algorithm is an odd hole, then output NOT ODD-HOLE-FREE and stop. Otherwise, merge the output with $\mathcal{L}_2$. Finally, in Step 3, we check whether every graph of $\mathcal{L}_2$ is basic. If this is the case, output ODD-HOLE-FREE. Otherwise output NOT ODD-HOLE-FREE.

The complexity of Step 1 is $O(|V(G)|^8)$ and there are $O(|V(G)|^2)$ graphs in $\mathcal{L}_1$. The 2-join decomposition algorithm, whose complexity is $O(|V(G)|^8)$, is applied $O(|V(G)|^2)$ times in Step 2 (since it is applied to every graph in $\mathcal{L}_1$), so the total complexity of Step 2 is $O(|V(G)|^{10})$. Given a graph G, the algorithms in [6] and [8] can test in time $O(|V(G)|^2)$ whether G is basic. Since there are $O(|V(G)|^3)$ graphs in $\mathcal{L}_2$, this implies that the complexity of Step 3 is $O(|V(G)|^5)$. Therefore the overall complexity of the algorithm is dominated by Step 2, which is $O(|V(G)|^{10})$. This proves 11.10. ∎

To apply this to test whether a graph is Berge, we need a version of 9.2 that will generate a set of subsets one of which is guaranteed to be a cleaner rather than just a near-cleaner. But it is easy to adapt 9.2 to do this (for each of the $O(|V(G)^5|$ sets X output by 9.2, remove at most three elements from it in all possible ways; of the $O(|V(G)^8|$ sets Y we generate, one is guaranteed to be a cleaner). The remainer of the application is just like in 10.1, applying 11.10 to all the graphs $G \setminus Y$, in place of applying 5.1 to the pairs G, X. The overall running time is $O(|V(G)|^{18})$.

12   Acknowledgements

References

[1] M. Chudnovsky, N. Robertson, P. Seymour and R. Thomas, "The strong perfect graph theorem", manuscript, June 2002.

[2] M. Conforti, G. Cornuéjols, A. Kapoor and K. Vušković, "Even-hole-free graphs, Part II: Recognition algorithm", *J. Graph Theory* 40 (2002), 238-266.

[3] M. Conforti, G. Cornuéjols and K. Vušković, "Decomposition of odd-hole-free graphs by double star cutsets and 2-joins", preprint (2001), to appear in the special issue of *Discrete Applied Mathematics* dedicated to the Brazilian Symposium on Graphs, Algorithms and Combinatorics, Fortaleza, March 17-19, 2001.

[4] M. Conforti and M.R. Rao, "Testing balancedness and perfection of linear matrices", *Mathematical Programming* 61 (1993), 1-18.

[5] G. Cornuéjols and W.H. Cunningham, "Compositions for perfect graphs", *Discrete Mathematics* 55 (1985), 245-254.

[6] P. G. H. Lehot, "An optimal algorithm to detect a line graph and output its root graph", *J. Assoc. Comput. Mach.* 21 (1974), 569-575.

[7] F. Roussel and P. Rubio, "About skew partitions in minimal imperfect graphs", *J. Combinatorial Theory, Ser. B* 83 (2001), 171-190.

[8] N. D. Roussopoulos, "A max$\{m, n\}$ algorithm for determining the graph H from its line graph G", *Informat. Process. Letters* 2 (1973), 108-112.